

# Register Allocation, ii

Liveness analysis & Graph coloring & their interplay

# Liveness analysis

Fixed point computation:

- Define which variables are set and which are used, for each instruction: **kill** & **gen** functions
- Specify how nearby instructions transmit live values around the program: **in** & **out** functions
- Iterate until nothing changes

**kill, gen : i  $\rightarrow$  register set**

kill(s) = {all assigned registers in stm s}

gen(s) = {all referenced registers in stm s}

# Gen & Kill

	<b>gen</b>	<b>kill</b>
1: <code>:f</code>	<code>()</code>	<code>()</code>
2: <code>(x2 &lt;- eax)</code>	<code>(eax)</code>	<code>(x2)</code>
3: <code>(x2 *= x2)</code>	<code>(x2)</code>	<code>(x2)</code>
4: <code>(2x2 &lt;- x2)</code>	<code>(x2)</code>	<code>(2x2)</code>
5: <code>(2x2 *= 2)</code>	<code>(2x2)</code>	<code>(2x2)</code>
6: <code>(3x &lt;- eax)</code>	<code>(eax)</code>	<code>(3x)</code>
7: <code>(3x *= 3)</code>	<code>(3x)</code>	<code>(3x)</code>
8: <code>(eax &lt;- 2x2)</code>	<code>(2x2)</code>	<code>(eax)</code>
9: <code>(eax += 3x)</code>	<code>(3x eax)</code>	<code>(eax)</code>
10: <code>(eax += 4)</code>	<code>(eax)</code>	<code>(eax)</code>
11: <code>(return)</code>	<code>(eax)</code>	<code>()</code>

**in, out : Nat  $\rightarrow$  register set**

$$\text{in}(n) = \text{gen}(n\text{-th-inst}) \cup (\text{out}(n) - \text{kill}(n\text{-th-inst}))$$

$$\text{out}(n) = \cup \{ \text{in}(m) \mid m \in \text{succ}(n) \}$$

**Nat** input is the number of an instruction in a function

Intuition behind the definitions: if  $x \in \mathbf{in}(n)$  then we know that  $x$  is live between instructions  $n-1$  and  $n$ .

Similarly, if  $x \in \mathbf{out}(n)$  then we know that  $x$  is live between instructions  $n$  and  $n+1$ .

**in, out : Nat → register set**

$$\text{in}(n) = \text{gen}(n\text{-th-inst}) \cup (\text{out}(n) - \text{kill}(n\text{-th-inst}))$$

$$\text{out}(n) = \cup \{ \text{in}(m) \mid m \in \text{succ}(n) \}$$

One possible solution: every variable is in **in**(n) and in **out**(n) for every n.

But that isn't super helpful... since we want to know what values don't need to be in registers at each point.

**in, out : Nat → register set**

$$\text{in}(n) = \text{gen}(n\text{-th-inst}) \cup (\text{out}(n) - \text{kill}(n\text{-th-inst}))$$

$$\text{out}(n) = \cup \{ \text{in}(m) \mid m \in \text{succ}(n) \}$$

Instead, compute **in** and **out** by iterating the equations until we reach a fixed point. Starting out with the assumption that they map everything to the empty set. Then repeatedly apply the right hand sides until nothing changes.

That result will satisfy the definitions of **in** and **out** but will it be the smallest solution?

# Liveness

	<b>in</b>	<b>out</b>
1: :f	()	()
2: (x2 <- eax)	()	()
3: (x2 *= x2)	()	()
4: (2x2 <- x2)	()	()
5: (2x2 *= 2)	()	()
6: (3x <- eax)	()	()
7: (3x *= 3)	()	()
8: (eax <- 2x2)	()	()
9: (eax += 3x)	()	()
10: (eax += 4)	()	()
11: (return)	()	()



# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	<code>()</code>	<code>()</code>
2: <code>(x2 &lt;- eax)</code>	<code>(eax)</code>	<code>()</code>
3: <code>(x2 *= x2)</code>	<code>(x2)</code>	<code>()</code>
4: <code>(2x2 &lt;- x2)</code>	<code>(x2)</code>	<code>()</code>
5: <code>(2x2 *= 2)</code>	<code>(2x2)</code>	<code>()</code>
6: <code>(3x &lt;- eax)</code>	<code>(eax)</code>	<code>()</code>
7: <code>(3x *= 3)</code>	<code>(3x)</code>	<code>()</code>
8: <code>(eax &lt;- 2x2)</code>	<code>(2x2)</code>	<code>()</code>
9: <code>(eax += 3x)</code>	<code>(3x eax)</code>	<code>()</code>
10: <code>(eax += 4)</code>	<code>(eax)</code>	<code>()</code>
11: <code>(return)</code>	<code>(eax)</code>	<code>()</code>

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	<code>()</code>	<code>(eax)</code>
2: <code>(x2 &lt;- eax)</code>	<code>(eax)</code>	<code>(x2)</code>
3: <code>(x2 *= x2)</code>	<code>(x2)</code>	<code>(x2)</code>
4: <code>(2x2 &lt;- x2)</code>	<code>(x2)</code>	<code>(2x2)</code>
5: <code>(2x2 *= 2)</code>	<code>(2x2)</code>	<code>(eax)</code>
6: <code>(3x &lt;- eax)</code>	<code>(eax)</code>	<code>(3x)</code>
7: <code>(3x *= 3)</code>	<code>(3x)</code>	<code>(2x2)</code>
8: <code>(eax &lt;- 2x2)</code>	<code>(2x2)</code>	<code>(3x eax)</code>
9: <code>(eax += 3x)</code>	<code>(3x eax)</code>	<code>(eax)</code>
10: <code>(eax += 4)</code>	<code>(eax)</code>	<code>(eax)</code>
11: <code>(return)</code>	<code>(eax)</code>	<code>()</code>

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(x2)
3: <code>(x2 *= x2)</code>	(x2)	(x2)
4: <code>(2x2 &lt;- x2)</code>	(x2)	(2x2)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(eax)
6: <code>(3x &lt;- eax)</code>	(eax)	(3x)
7: <code>(3x *= 3)</code>	(2x2 3x)	(2x2)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x)	(3x eax)
9: <code>(eax += 3x)</code>	(3x eax)	(eax)
10: <code>(eax += 4)</code>	(eax)	(eax)
11: <code>(return)</code>	(eax)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(x2)
3: <code>(x2 *= x2)</code>	(x2)	(x2)
4: <code>(2x2 &lt;- x2)</code>	(x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(eax)
6: <code>(3x &lt;- eax)</code>	(eax)	(2x2 3x)
7: <code>(3x *= 3)</code>	(2x2 3x)	(2x2 3x)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x)	(3x eax)
9: <code>(eax += 3x)</code>	(3x eax)	(eax)
10: <code>(eax += 4)</code>	(eax)	(eax)
11: <code>(return)</code>	(eax)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(x2)
3: <code>(x2 *= x2)</code>	(x2)	(x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(eax)
6: <code>(3x &lt;- eax)</code>	(2x2 eax)	(2x2 3x)
7: <code>(3x *= 3)</code>	(2x2 3x)	(2x2 3x)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x)	(3x eax)
9: <code>(eax += 3x)</code>	(3x eax)	(eax)
10: <code>(eax += 4)</code>	(eax)	(eax)
11: <code>(return)</code>	(eax)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(x2)
3: <code>(x2 *= x2)</code>	(x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(2x2 eax)
6: <code>(3x &lt;- eax)</code>	(2x2 eax)	(2x2 3x)
7: <code>(3x *= 3)</code>	(2x2 3x)	(2x2 3x)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x)	(3x eax)
9: <code>(eax += 3x)</code>	(3x eax)	(eax)
10: <code>(eax += 4)</code>	(eax)	(eax)
11: <code>(return)</code>	(eax)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(x2)
3: <code>(x2 *= x2)</code>	(eax x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(2x2 eax)
6: <code>(3x &lt;- eax)</code>	(2x2 eax)	(2x2 3x)
7: <code>(3x *= 3)</code>	(2x2 3x)	(2x2 3x)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x)	(3x eax)
9: <code>(eax += 3x)</code>	(3x eax)	(eax)
10: <code>(eax += 4)</code>	(eax)	(eax)
11: <code>(return)</code>	(eax)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(eax x2)
3: <code>(x2 *= x2)</code>	(eax x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(2x2 eax)
6: <code>(3x &lt;- eax)</code>	(2x2 eax)	(2x2 3x)
7: <code>(3x *= 3)</code>	(2x2 3x)	(2x2 3x)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x)	(3x eax)
9: <code>(eax += 3x)</code>	(3x eax)	(eax)
10: <code>(eax += 4)</code>	(eax)	(eax)
11: <code>(return)</code>	(eax)	()



# Graph coloring

Reduce register allocation to the *graph coloring* problem

- Nodes represent variables
- Edges connect nodes that cannot share a register
- Colors represent registers

If we can color the graph (where no adjacent nodes share a color) in  $N$  colors, then we can compile this program to use  $N$  registers

# Graph coloring

Build interference graph from the liveness information

- For each instruction:
  - Two variables live at the same time interfere with each other
  - Killed variables interfere with variables in the out set
  - Except that the variables **x** and **y** do not interfere if the instruction was **(x ← y)**
- All real registers interfere with each other

# Graph coloring

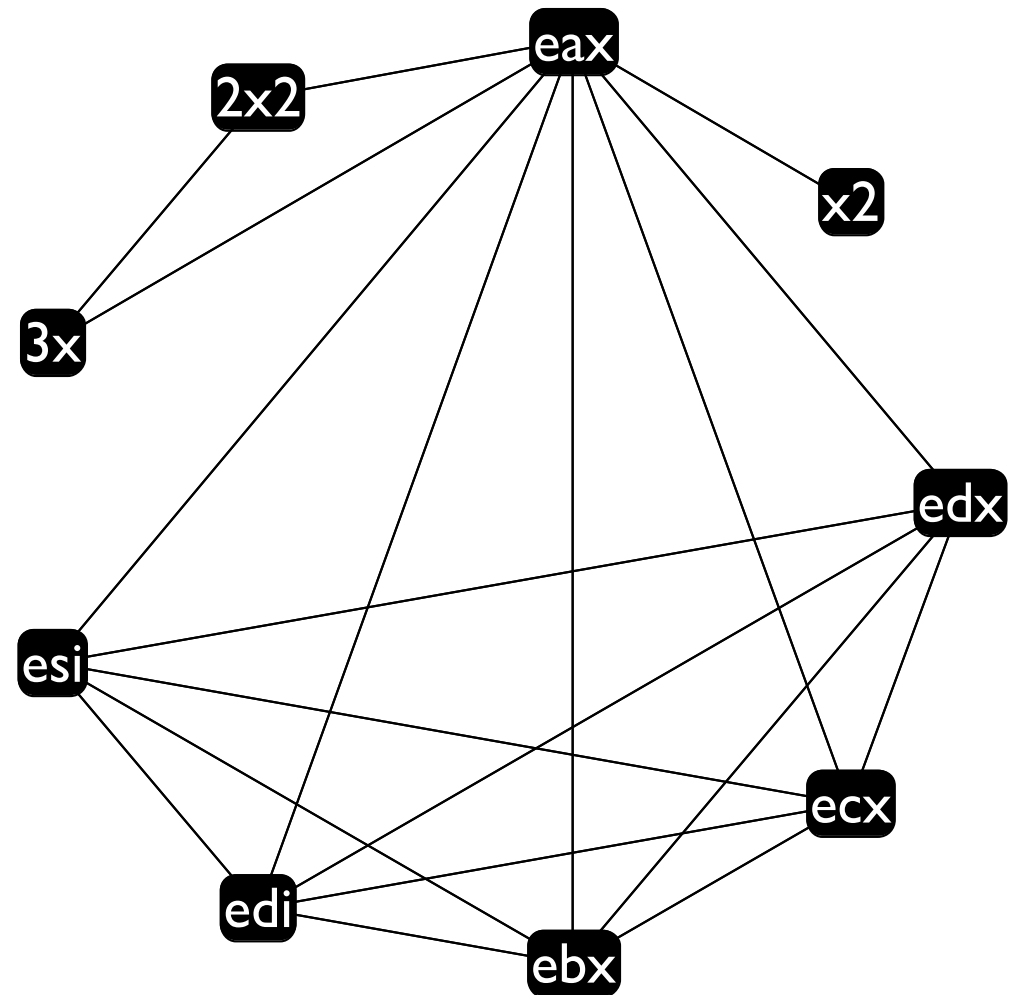
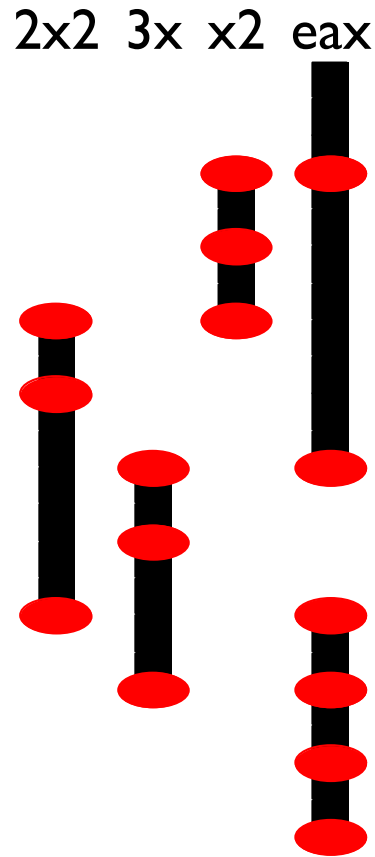
What counts as live “at the same time”?

- We could say that two variables are live at the same time if they appear in either the ‘in’ or the ‘out’ set together for any instruction
- But, not counting instructions that have no predecessors, every variable pair that is in an ‘in’ set, is also in an ‘out’ set
- Also, the only instruction that has no predecessor that actually gets run is the first one

So, live at the same time means appearing together in an ‘out’ set, or appearing together in the first instruction’s ‘in’ set

# Liveness $\Rightarrow$ Interference graph

```
: f
(x2 <- eax)
(x2 *= x2)
(2x2 <- x2)
(2x2 *= 2)
(3x <- eax)
(3x *= 3)
(eax <- 2x2)
(eax += 3x)
(eax += 4)
(return)
```



# Check yourself

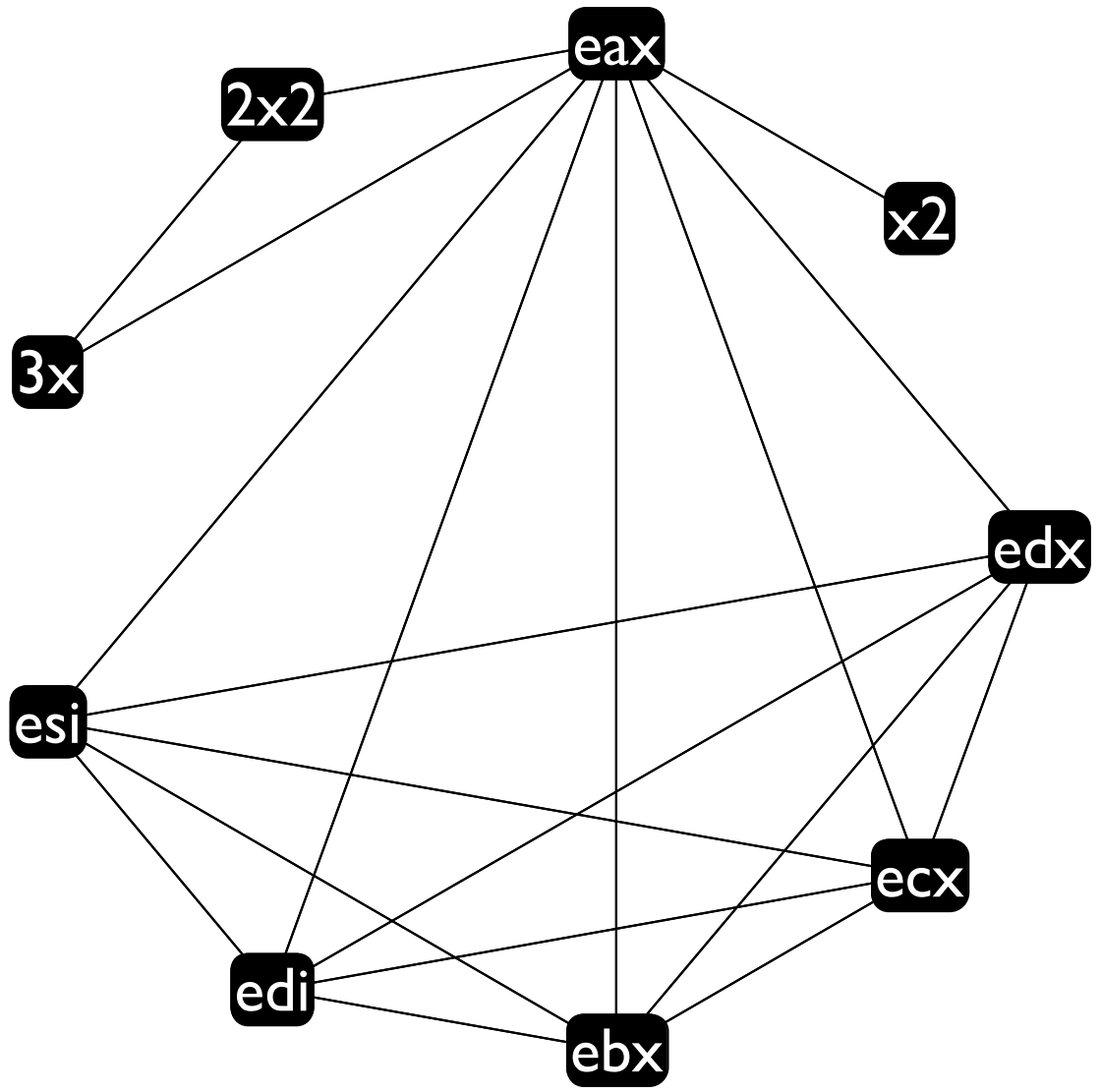
For each pair of  $x^2$ ,  $2x^2$ ,  $3x$ , and  $eax$ :

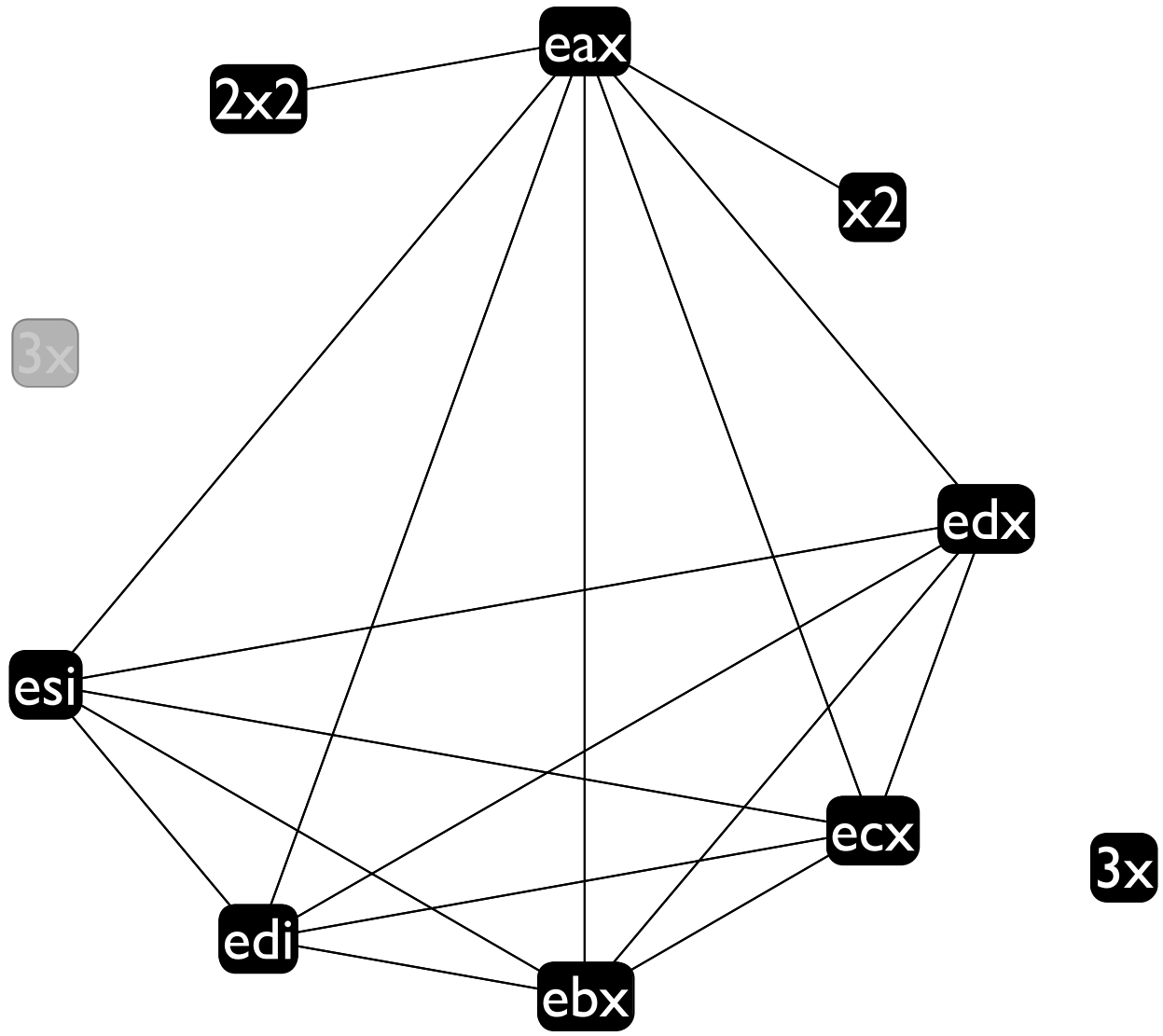
- do they interfere?;
- should they?

# Graph coloring

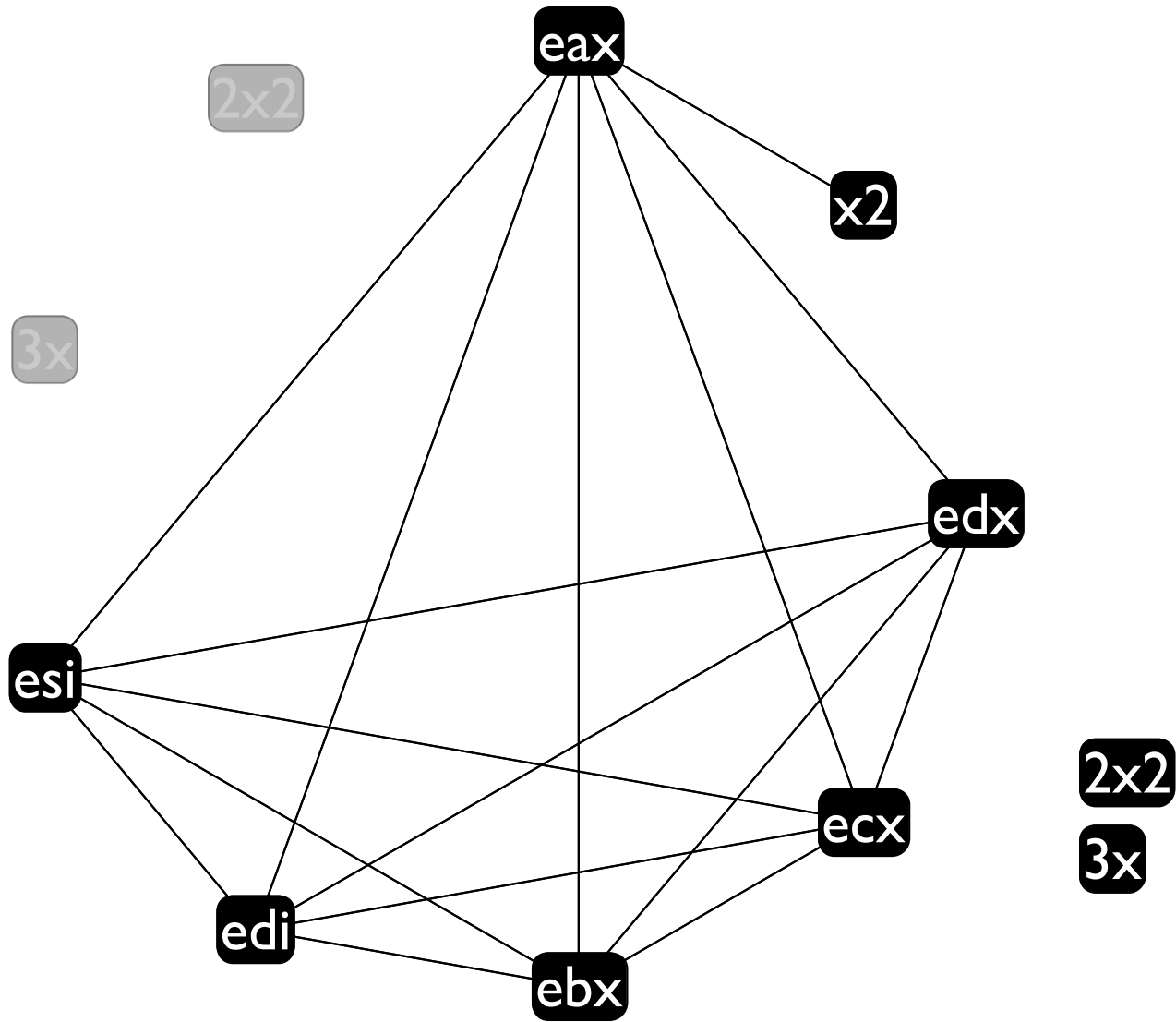
Algorithm:

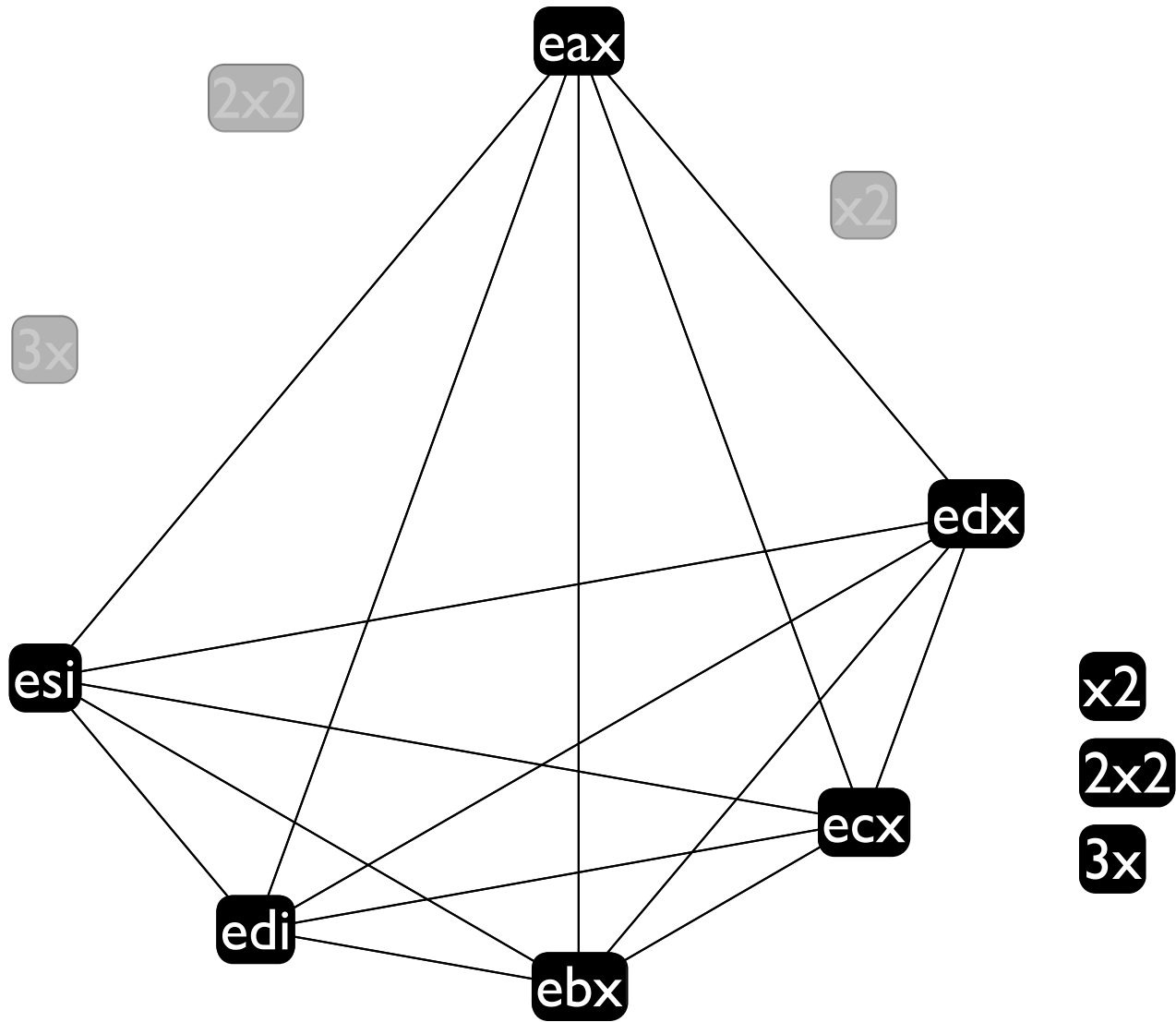
- Repeatedly select a node and remove it from the graph, putting it onto a stack
- When the graph is empty, rebuild it, putting a new color on each node as it comes back into the graph, making sure no adjacent nodes have the same color
- If there are not enough colors, the algorithm fails (spilling comes in here)
- Make sure real registers come out of the graph last so they get the first 6 colors

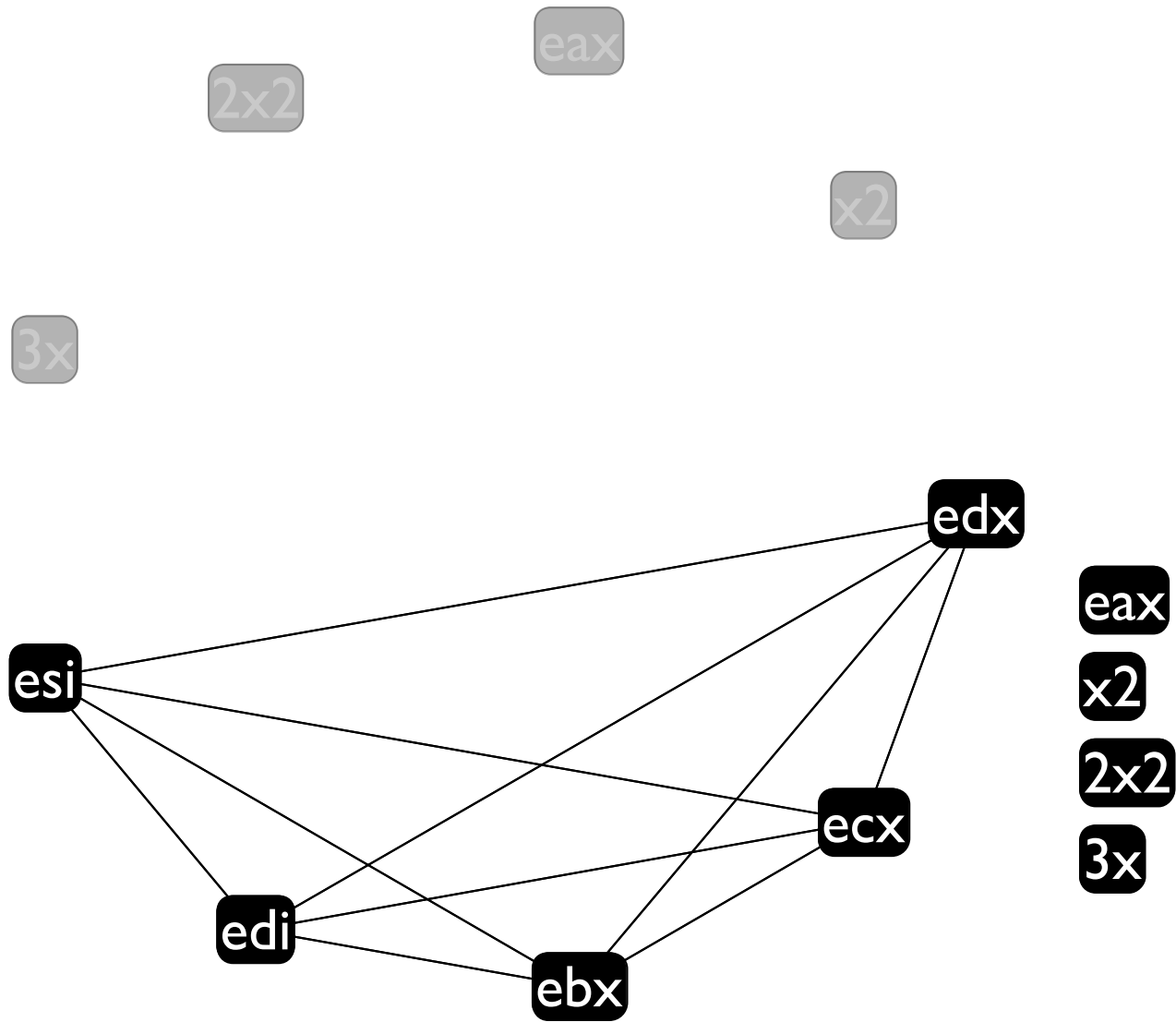


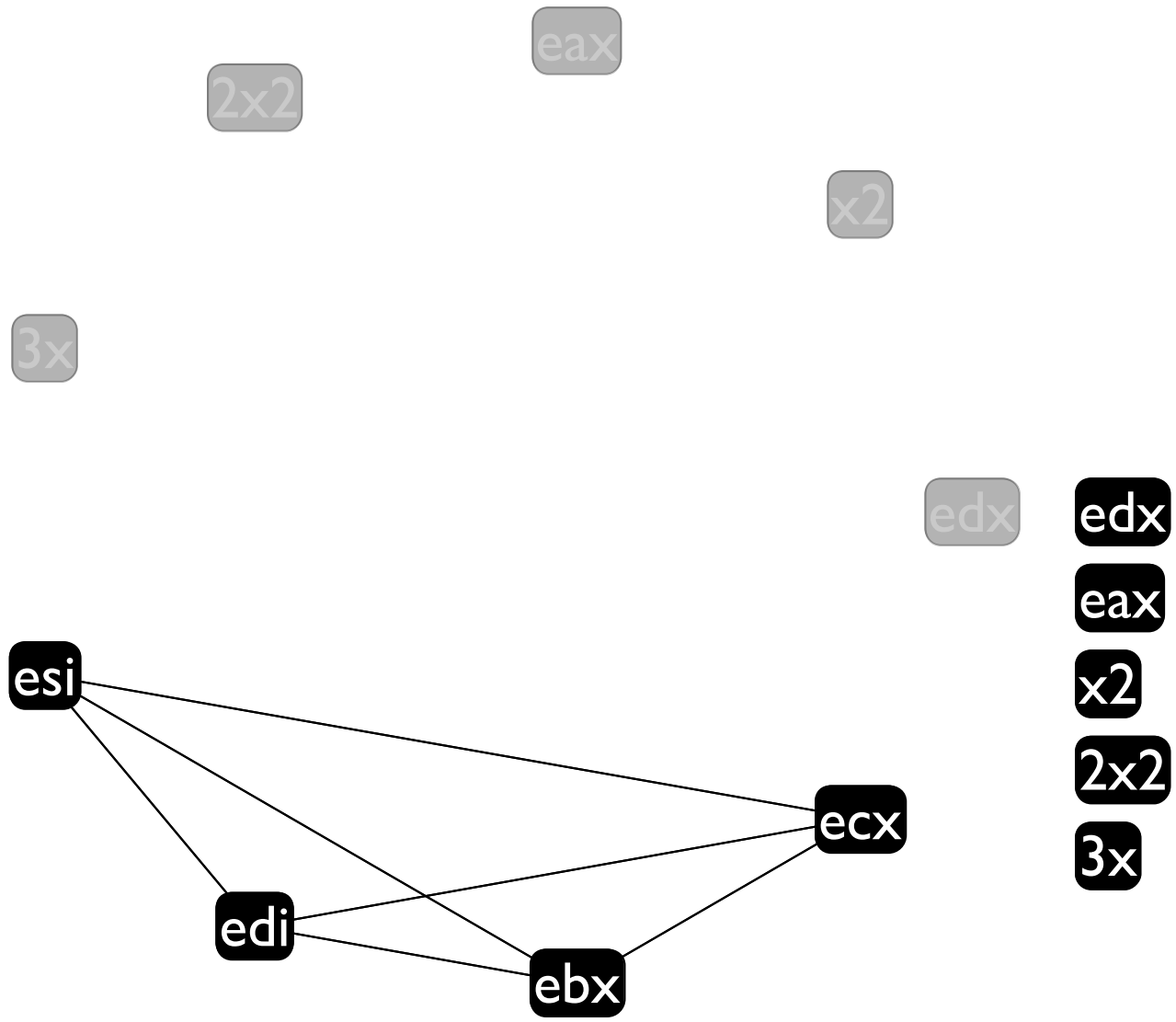


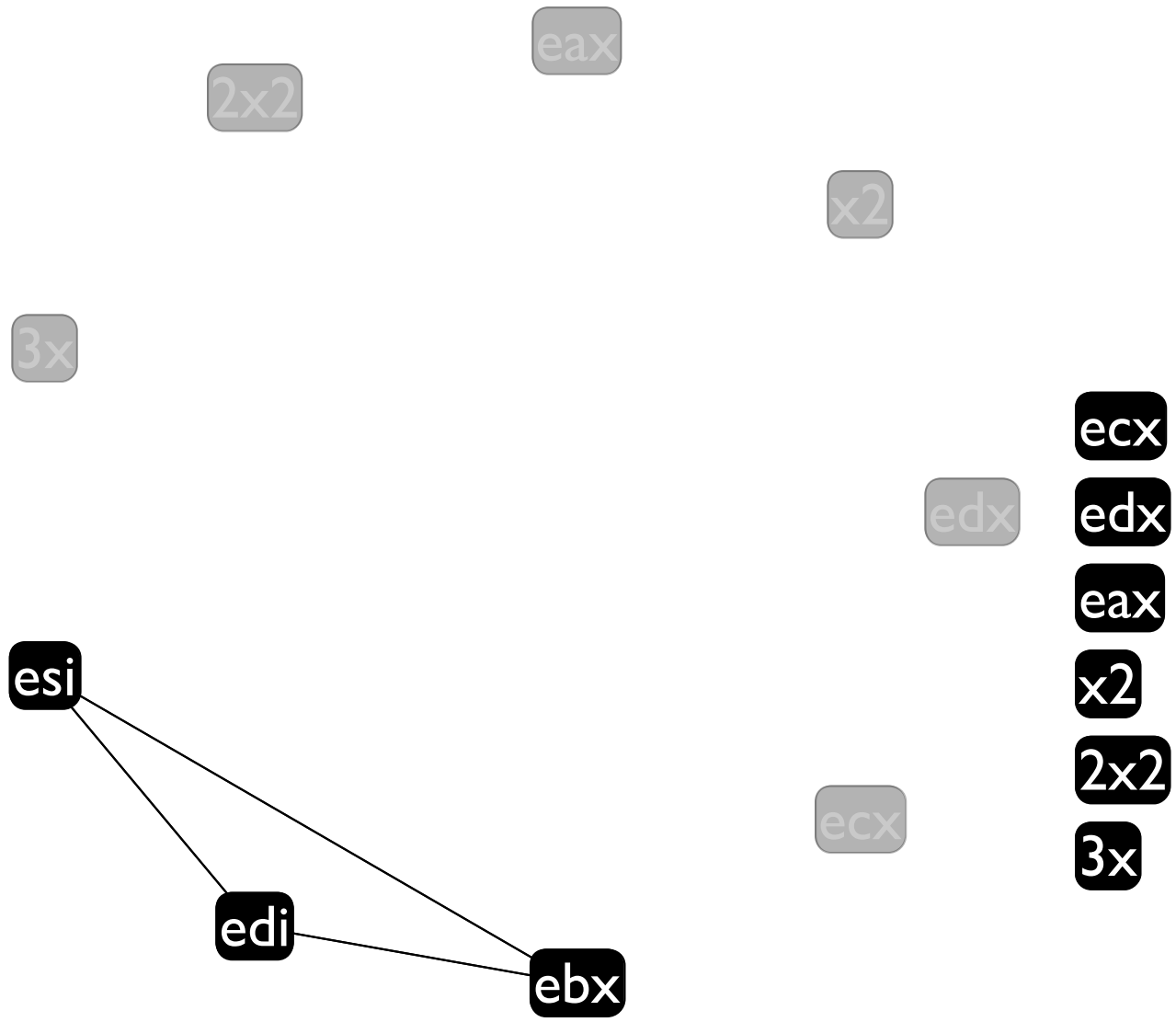


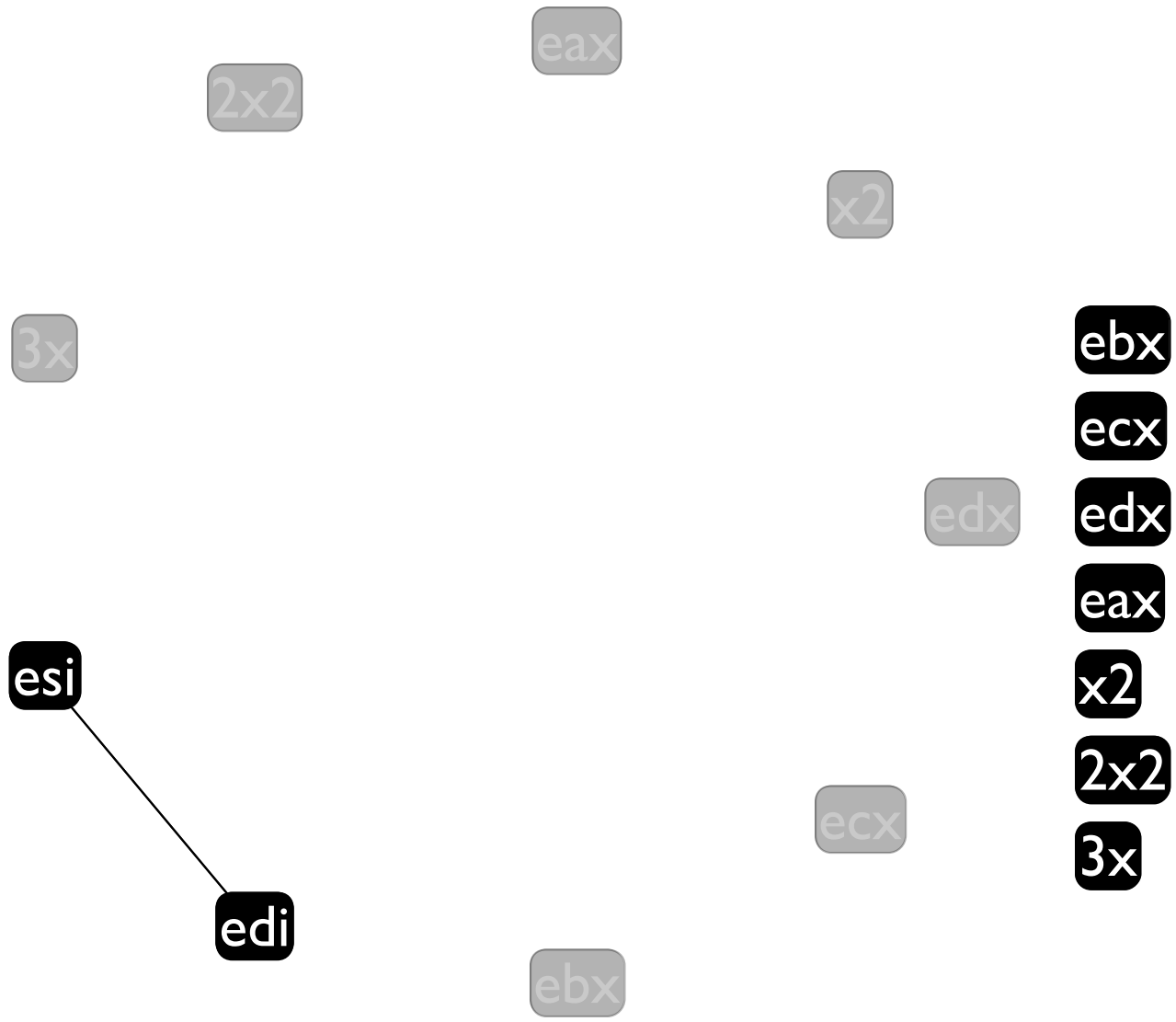


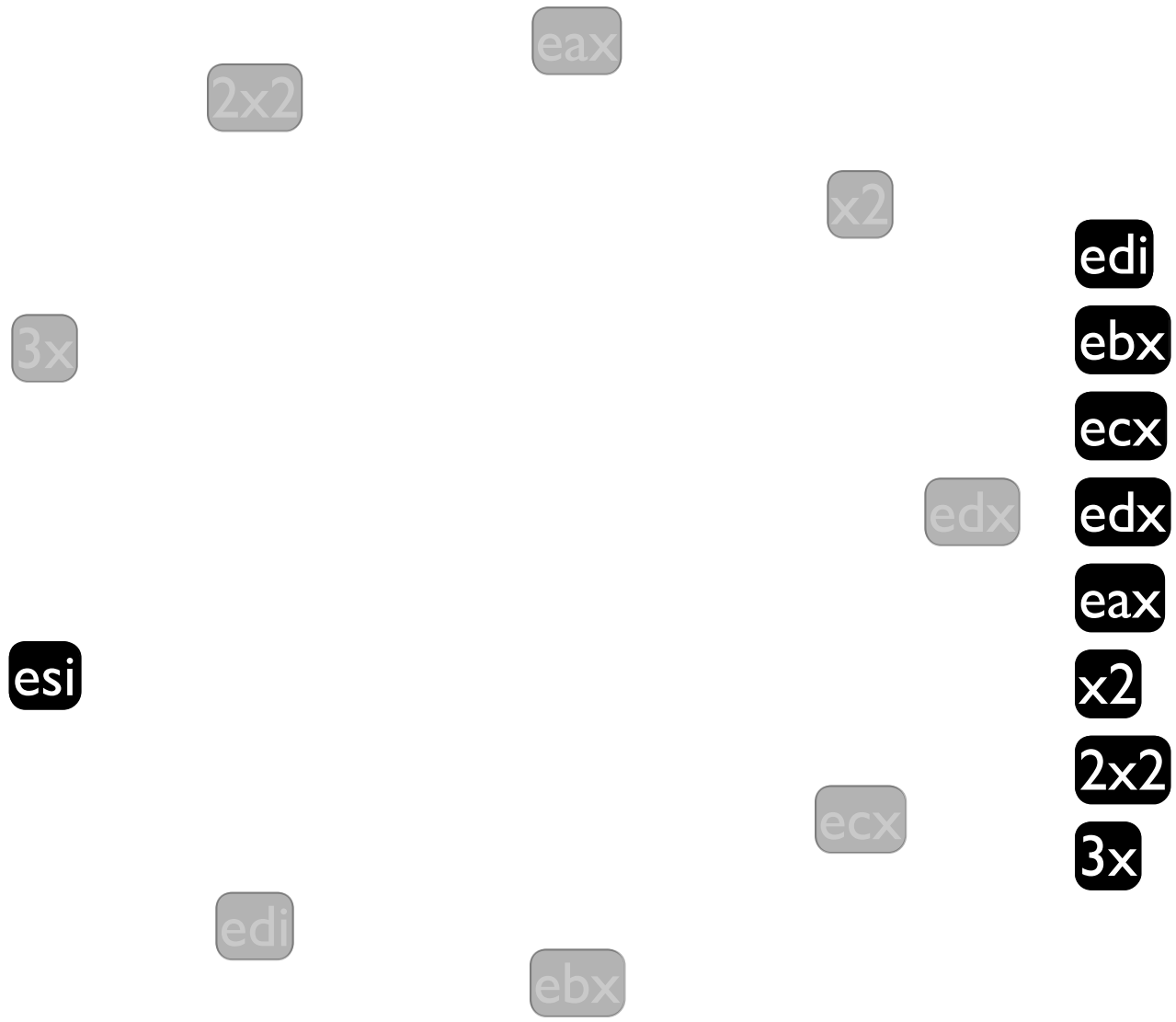


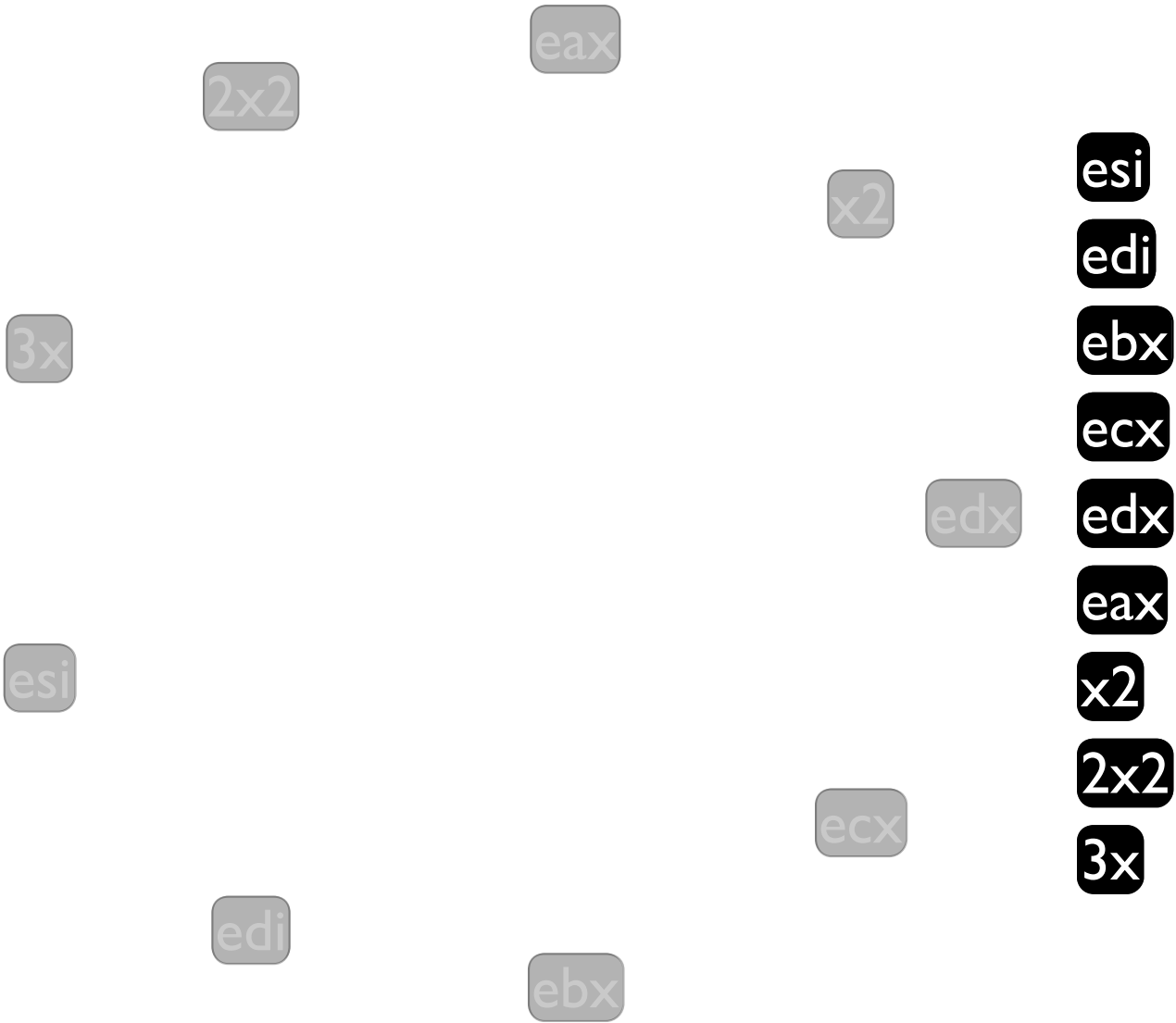




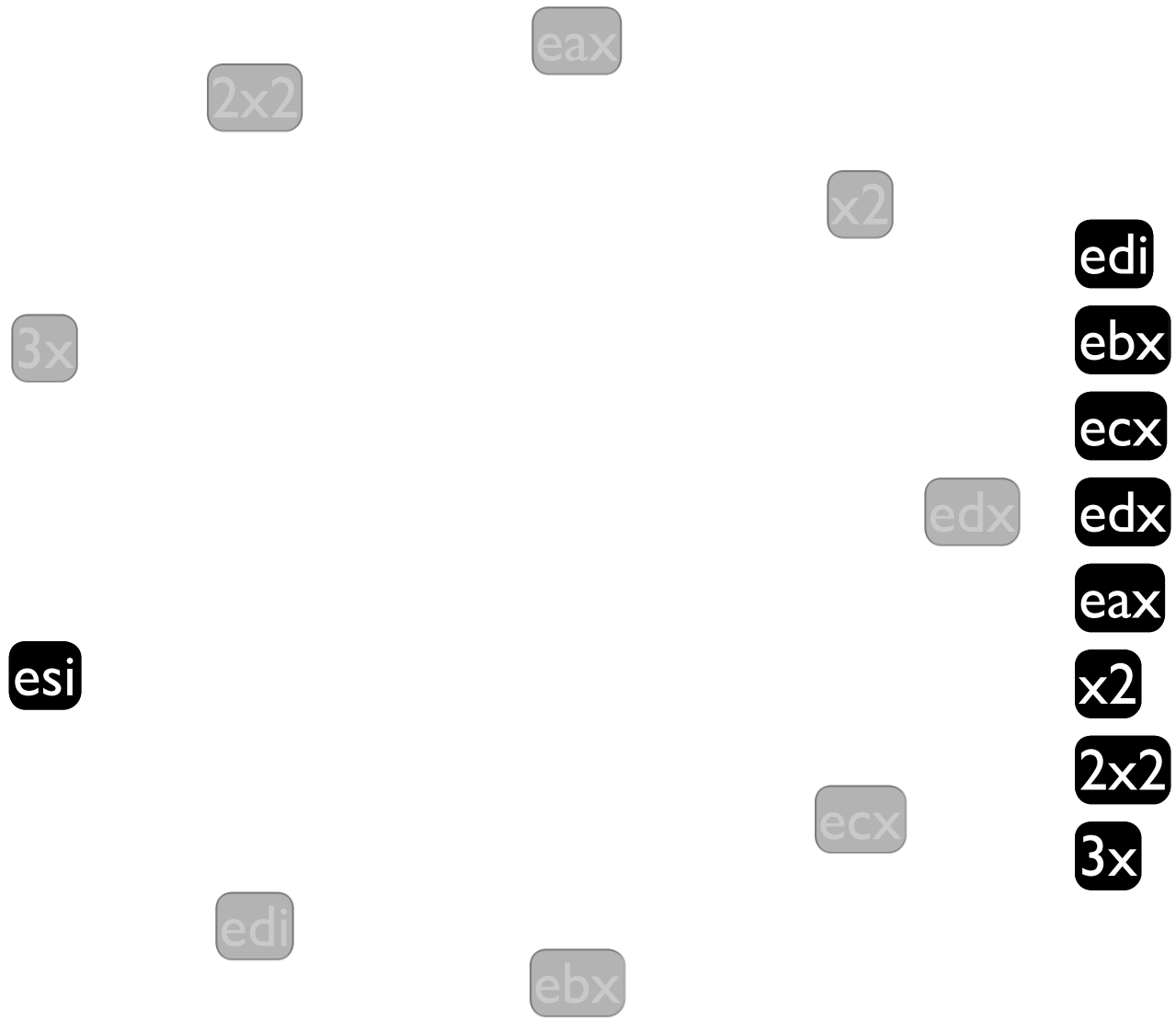


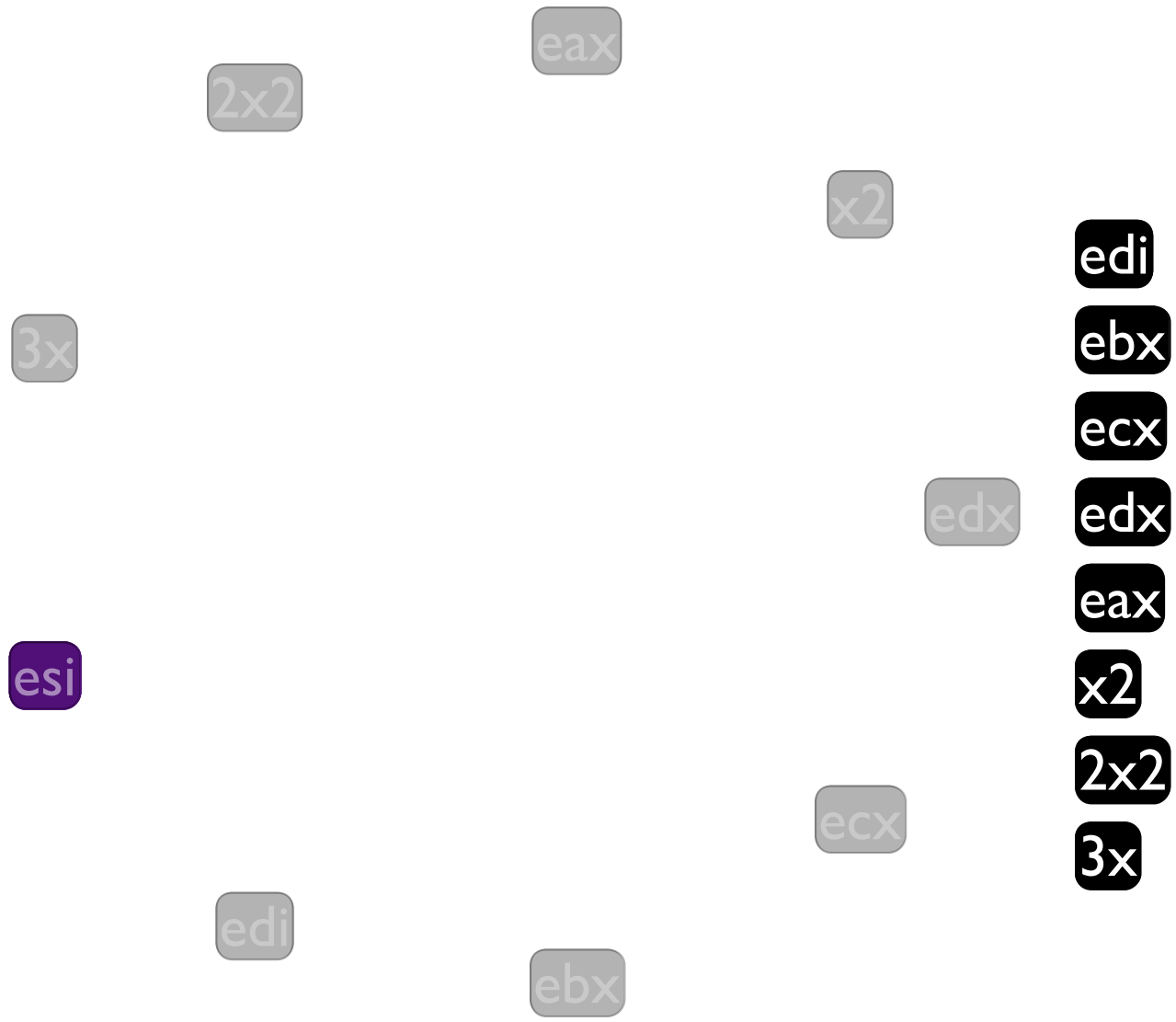




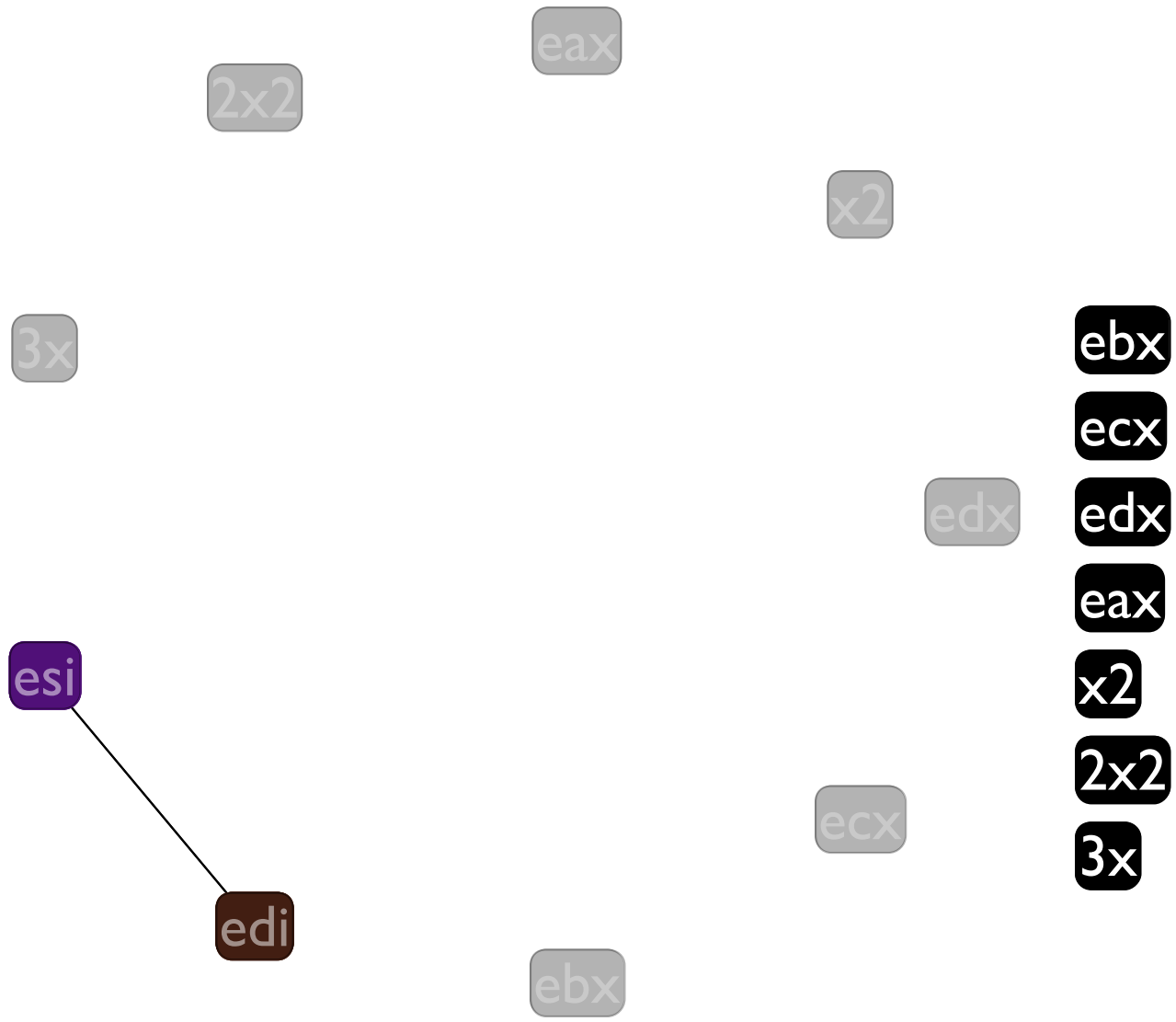


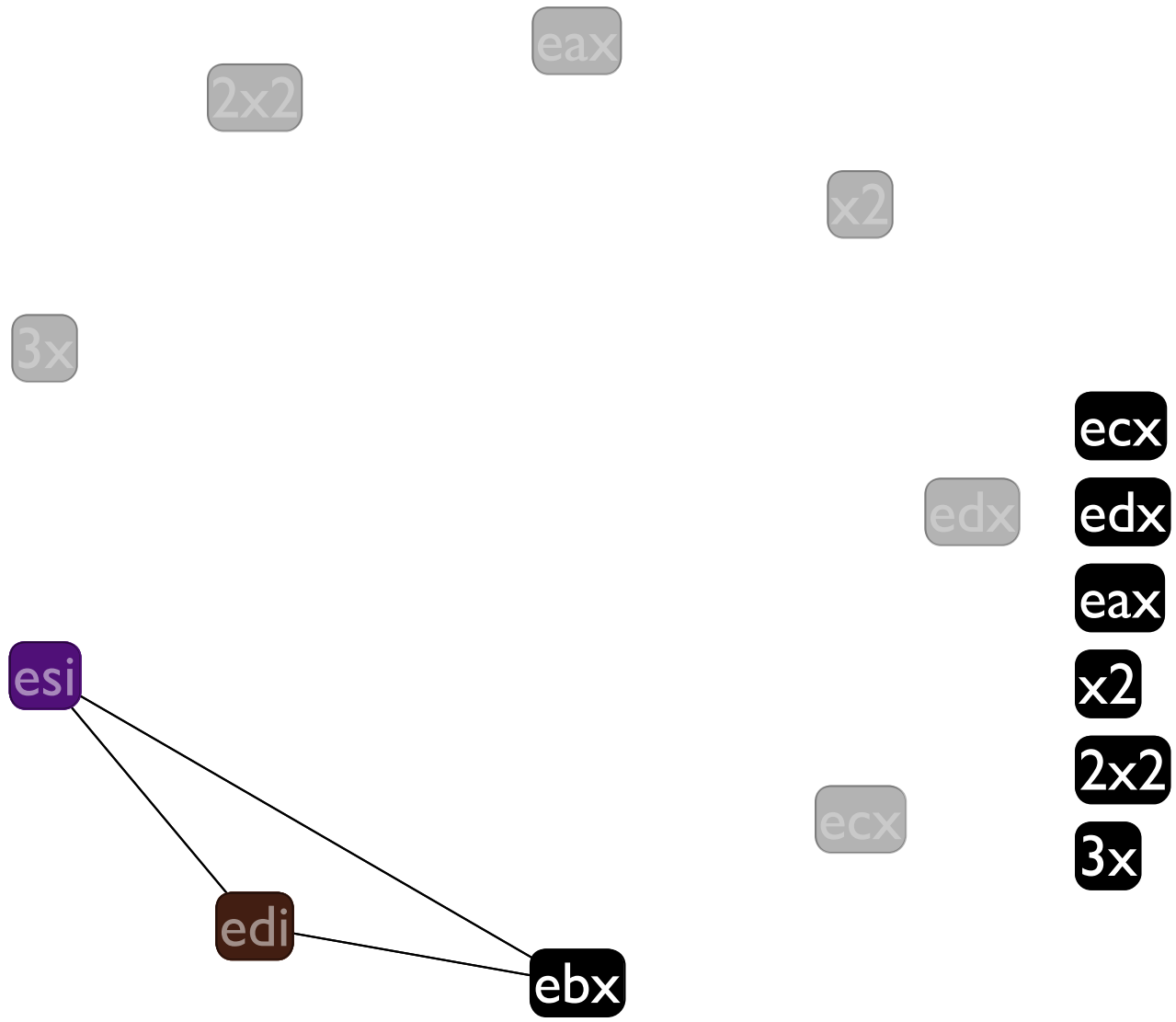


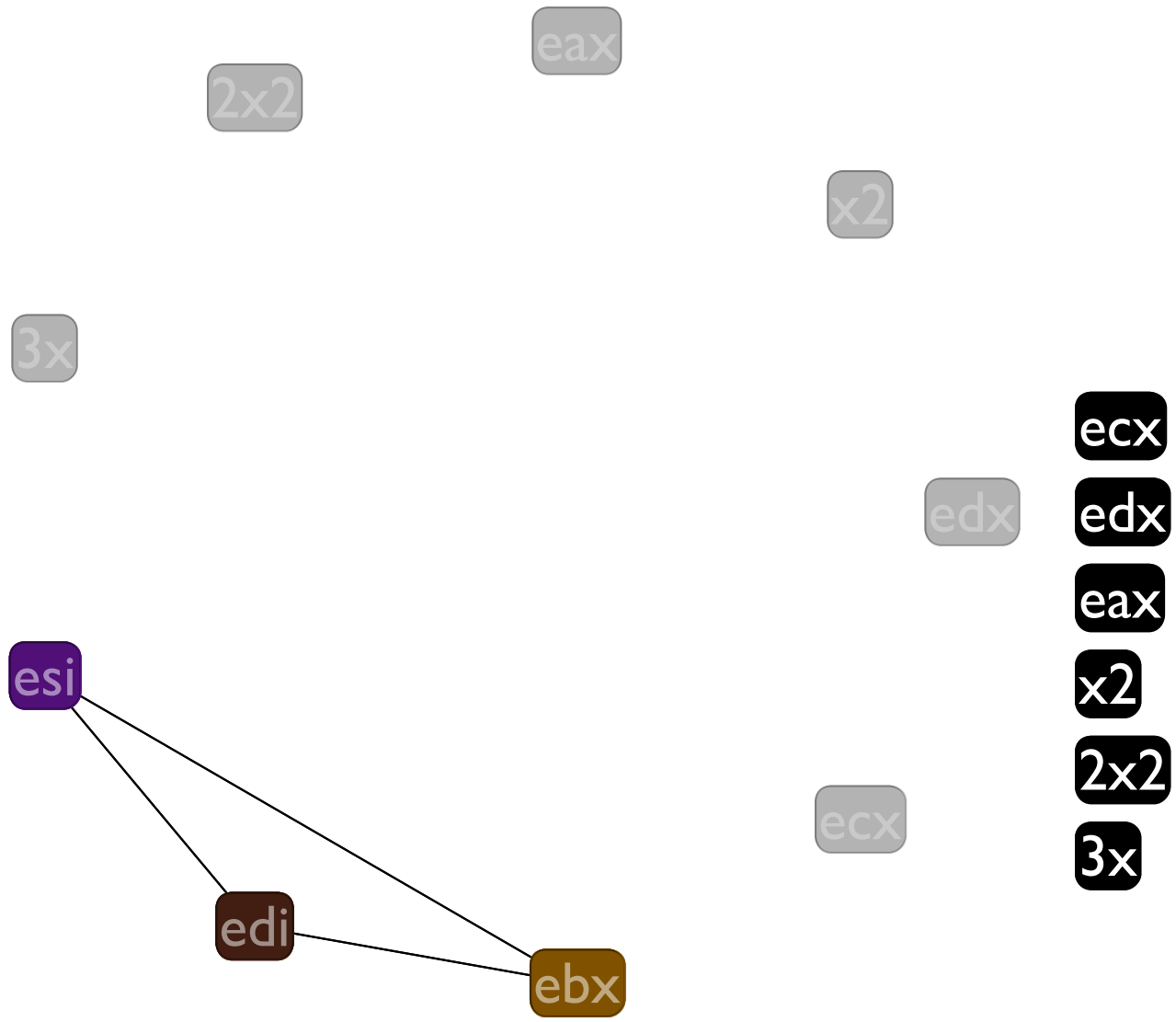


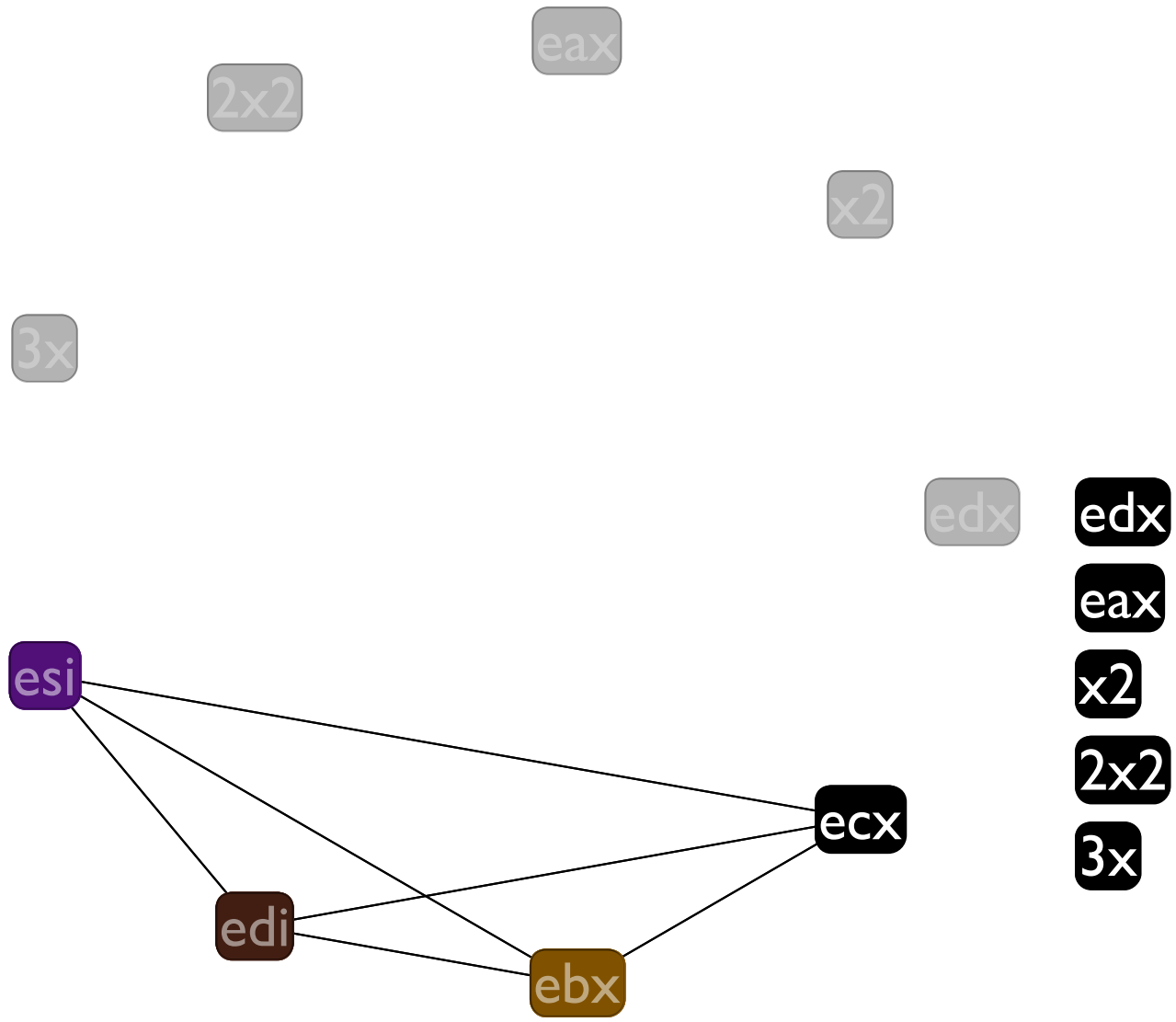


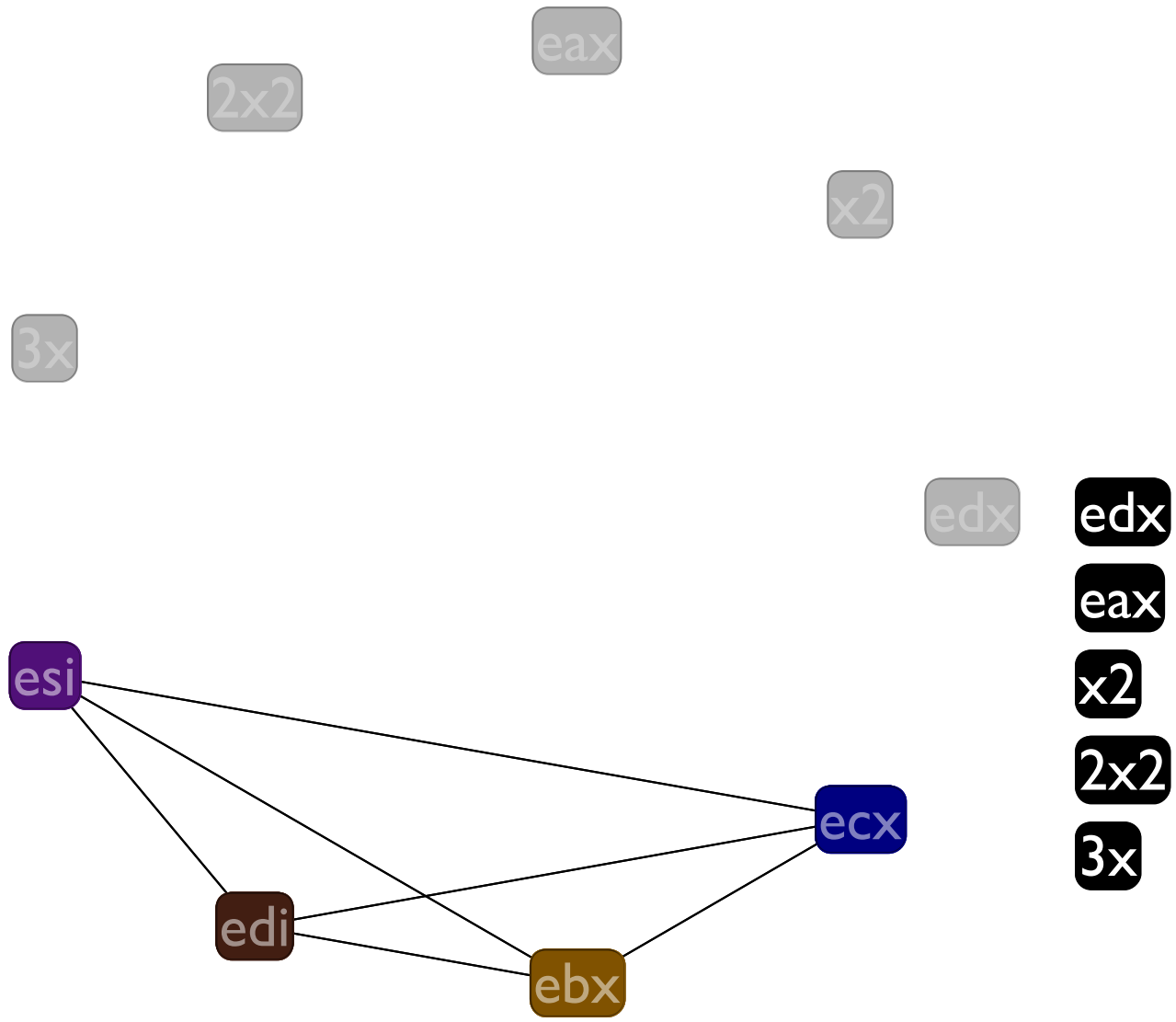




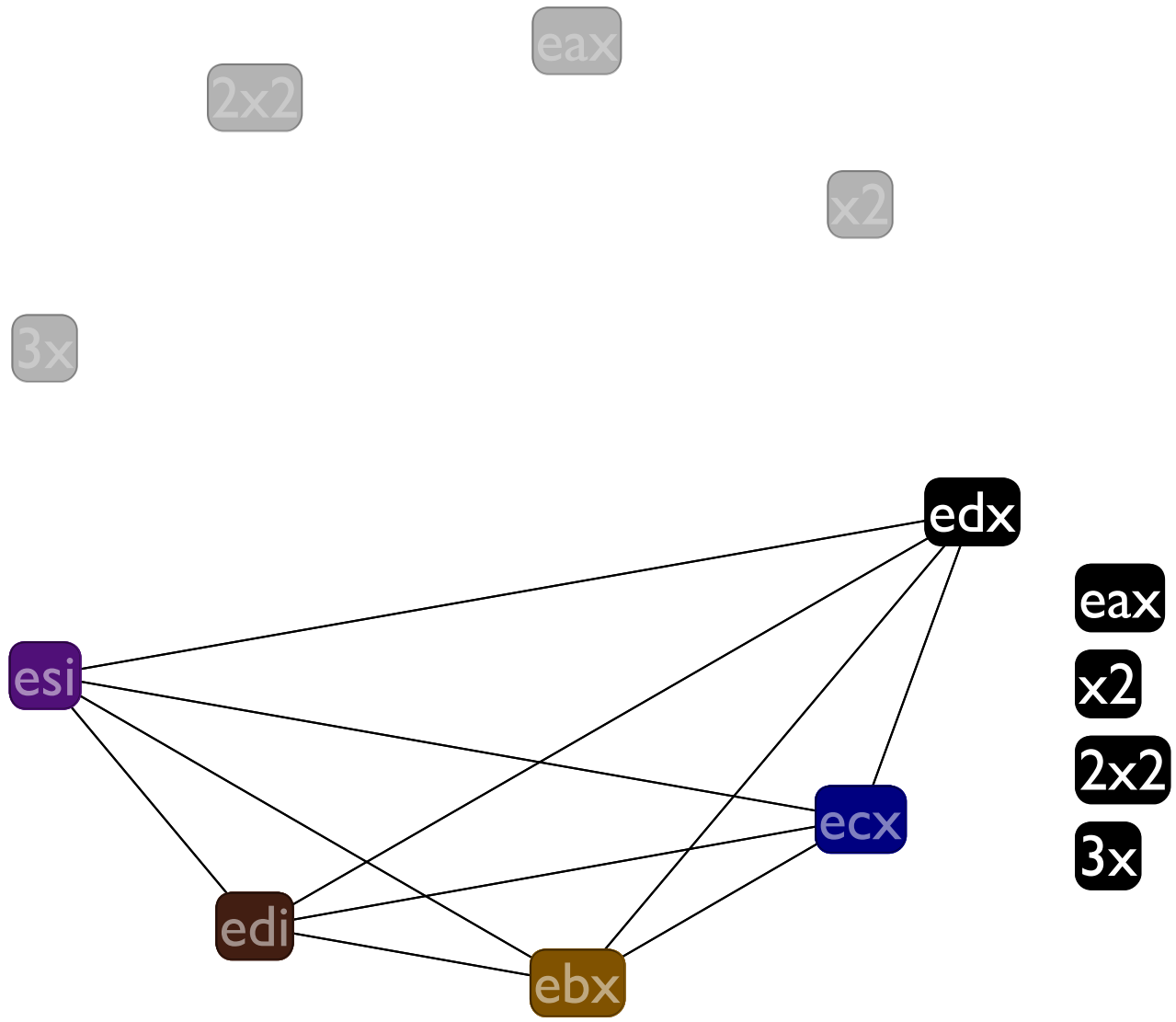


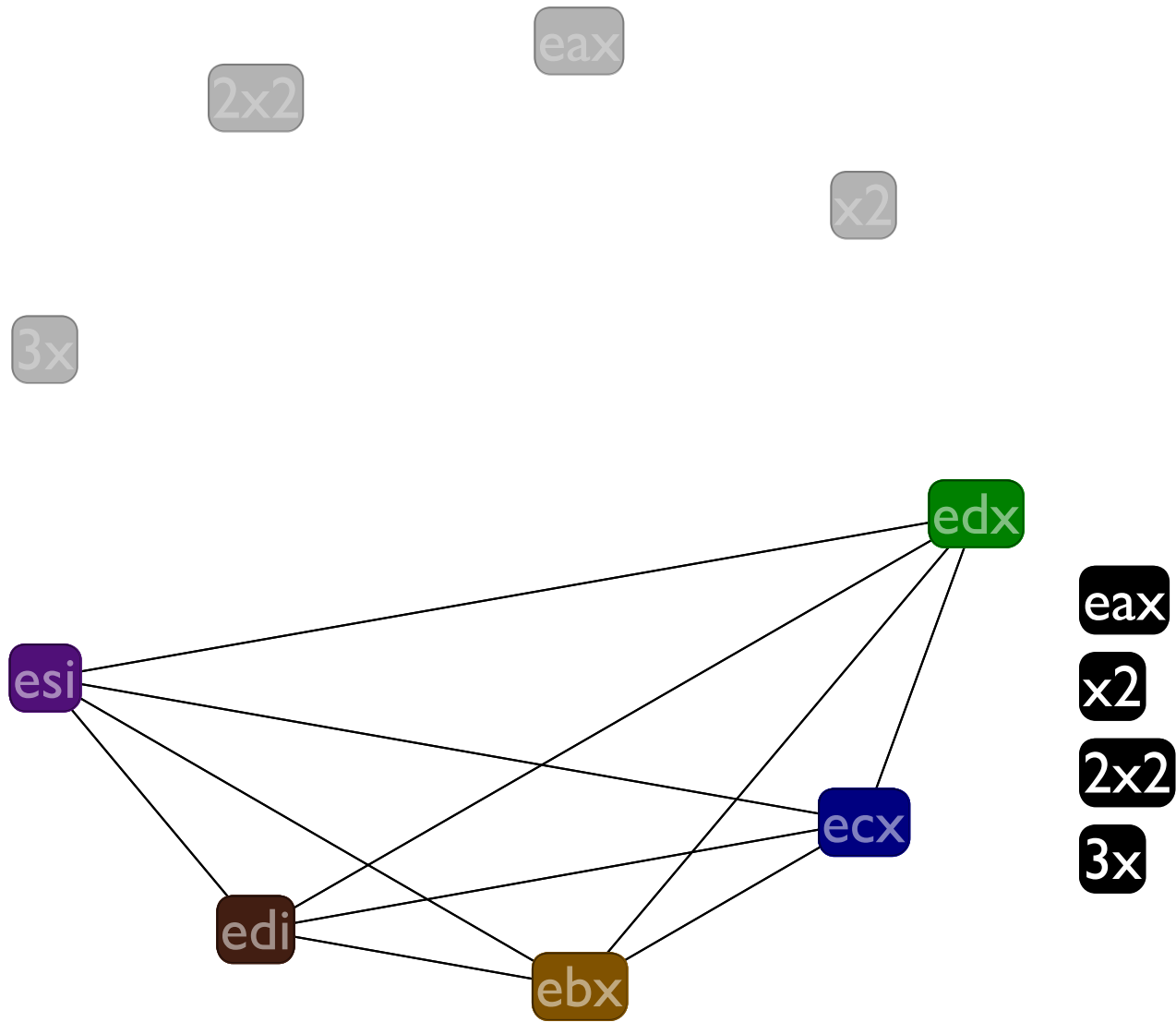


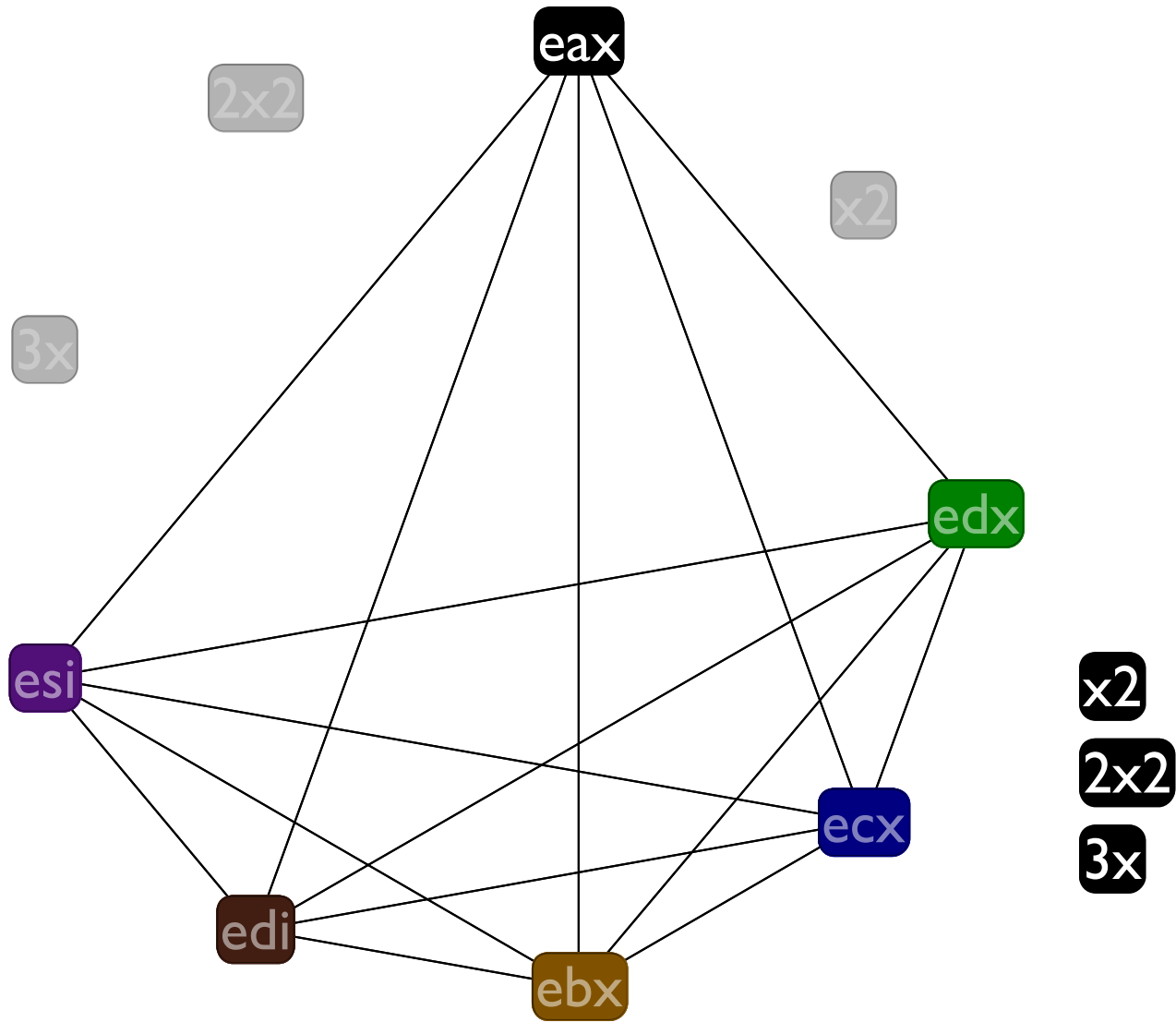


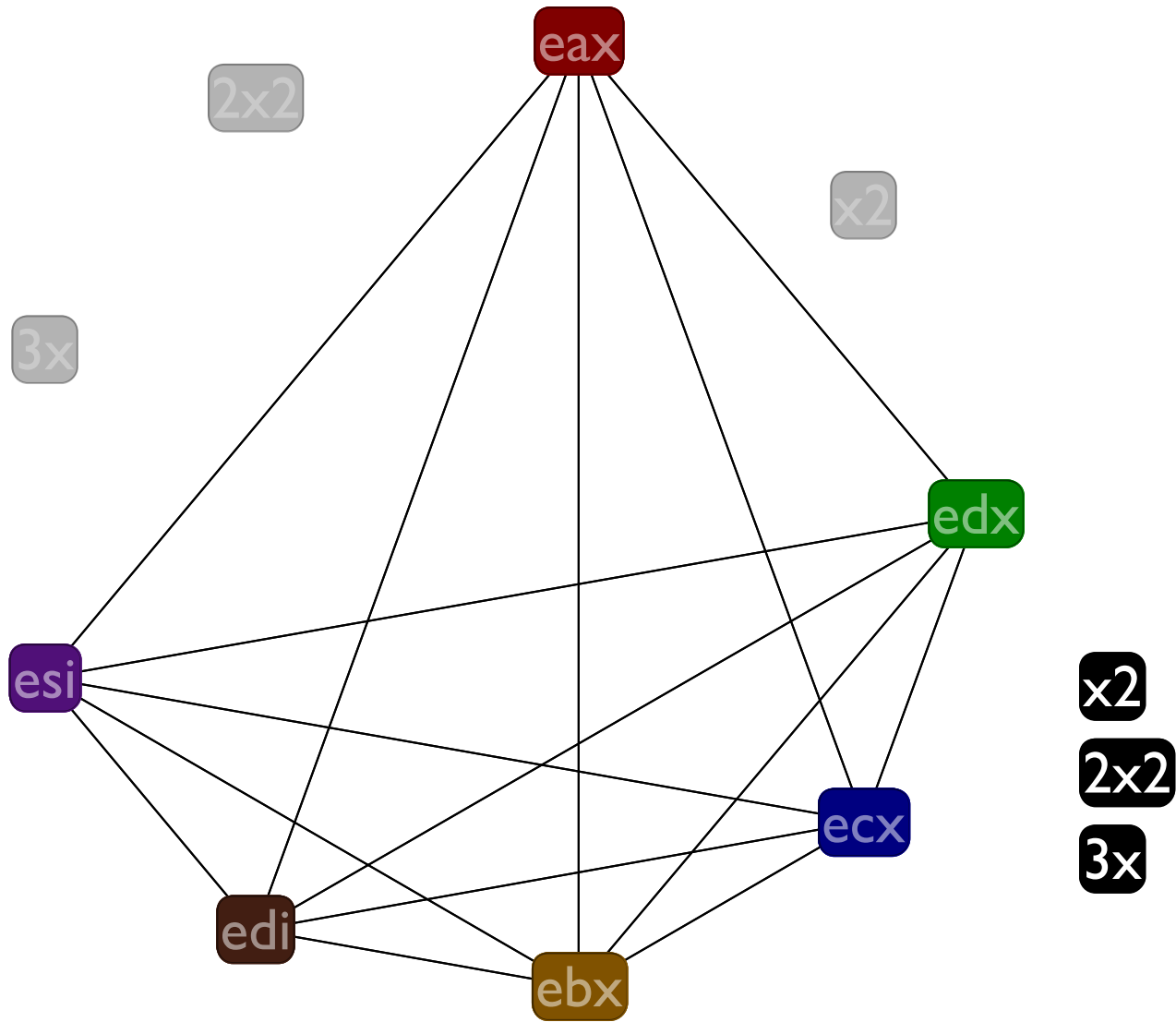


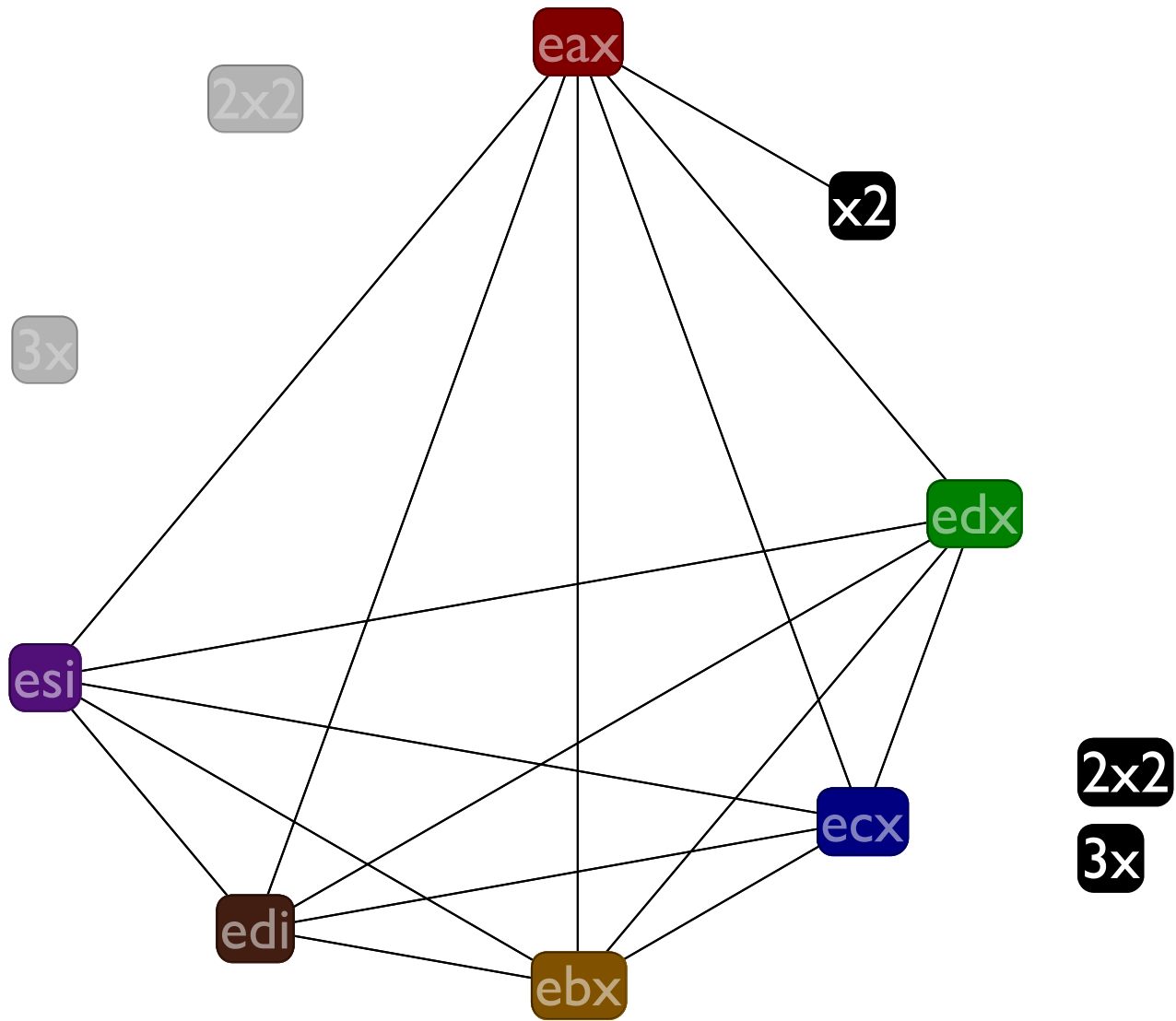


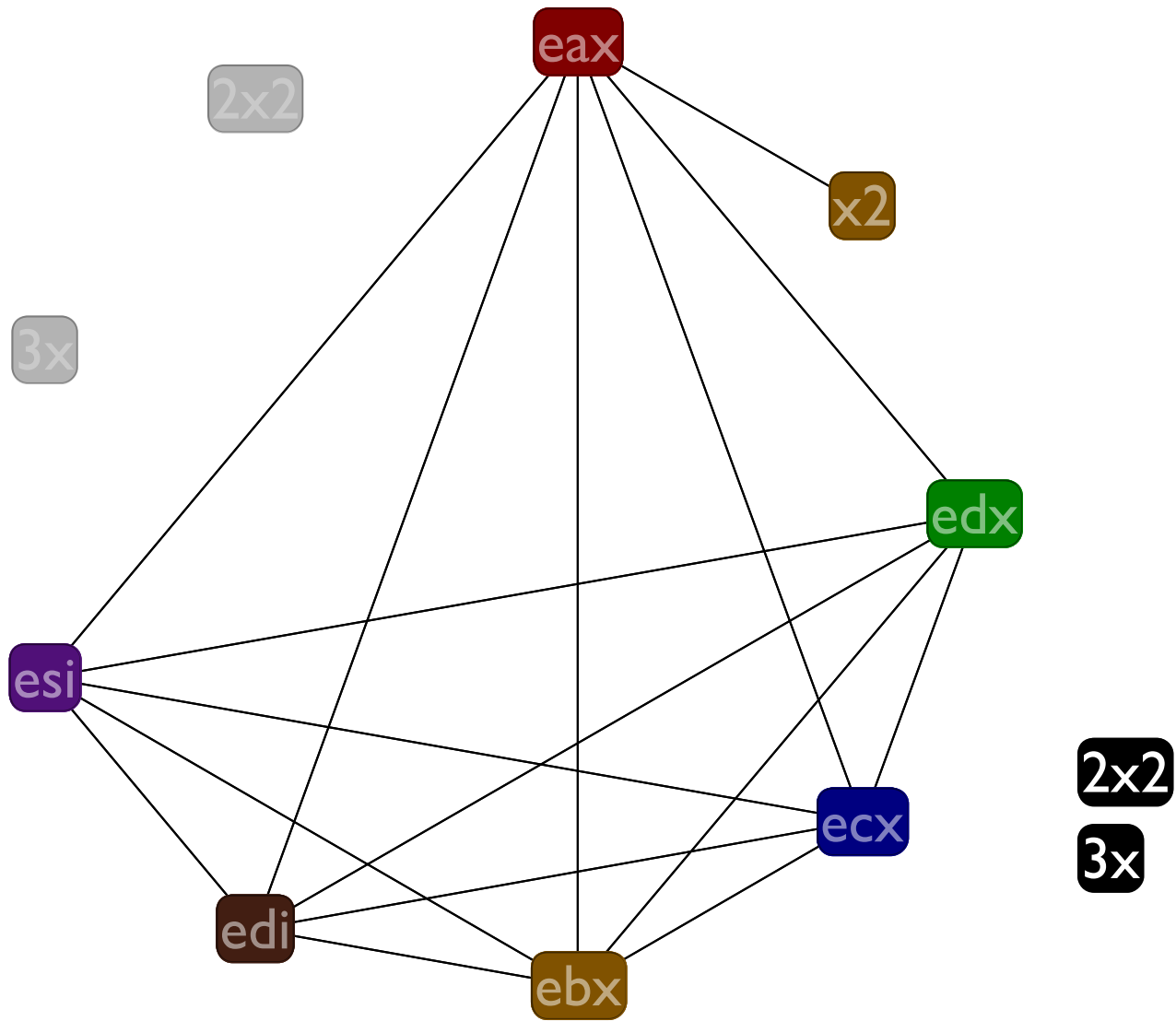


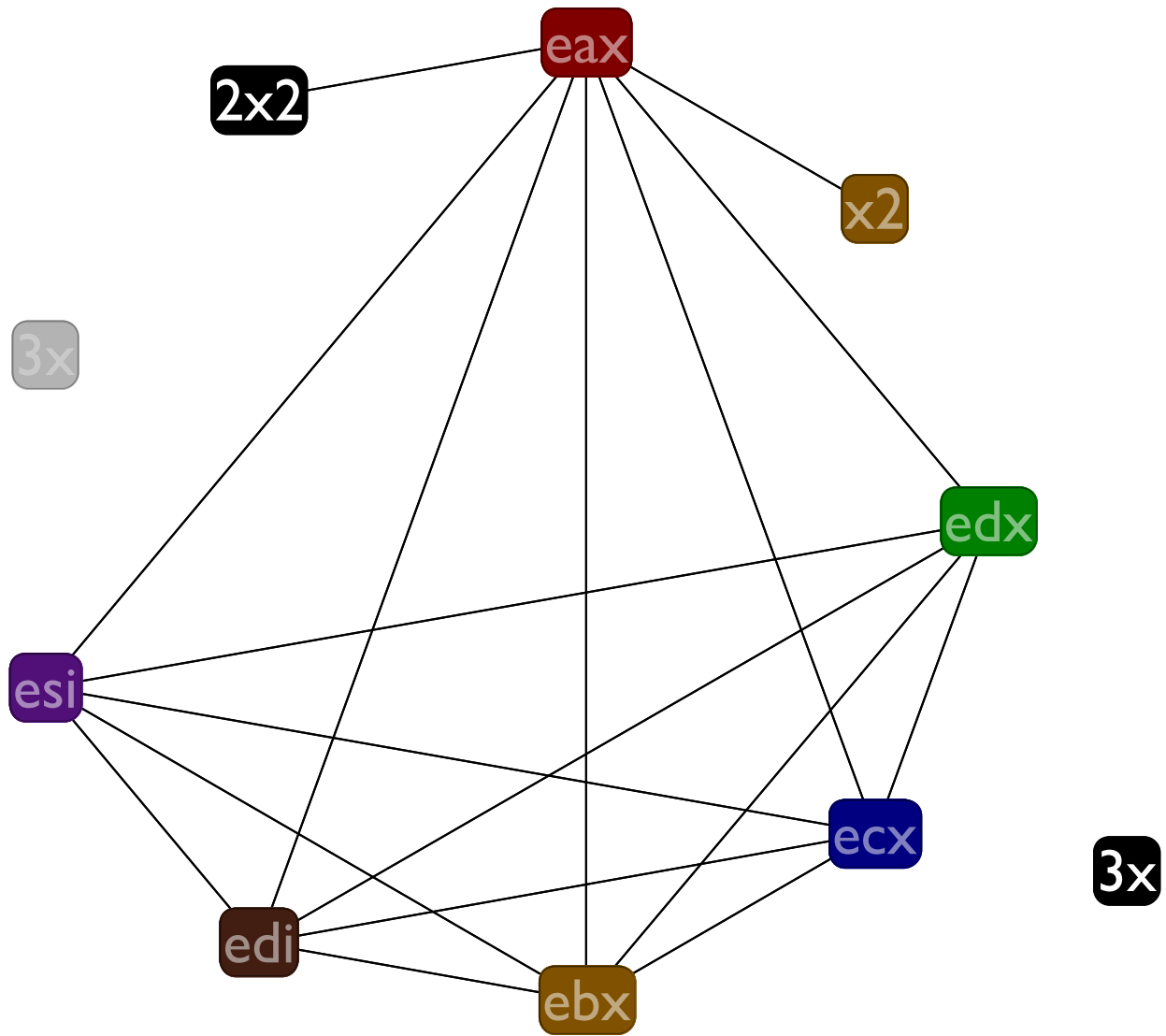


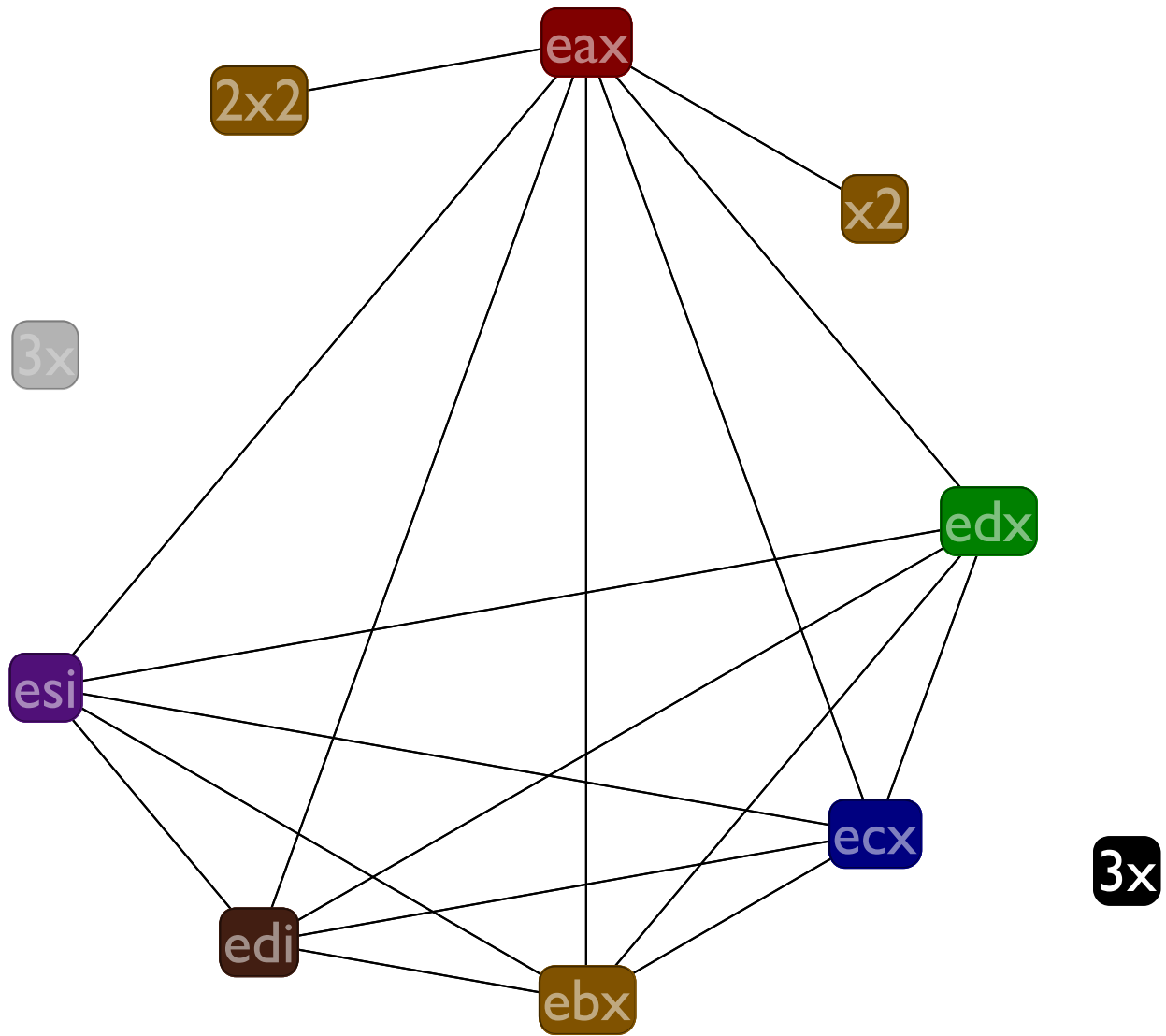




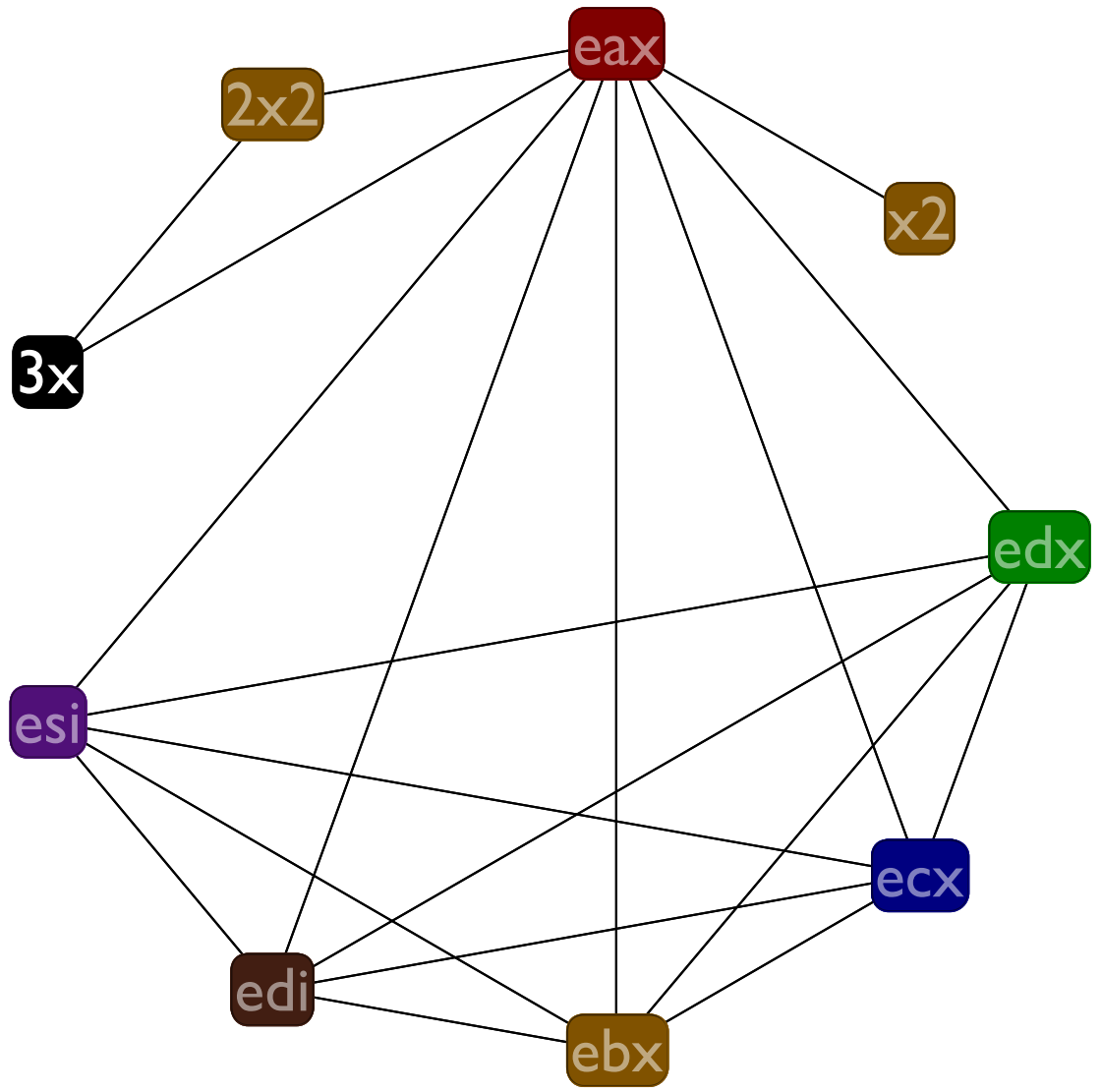


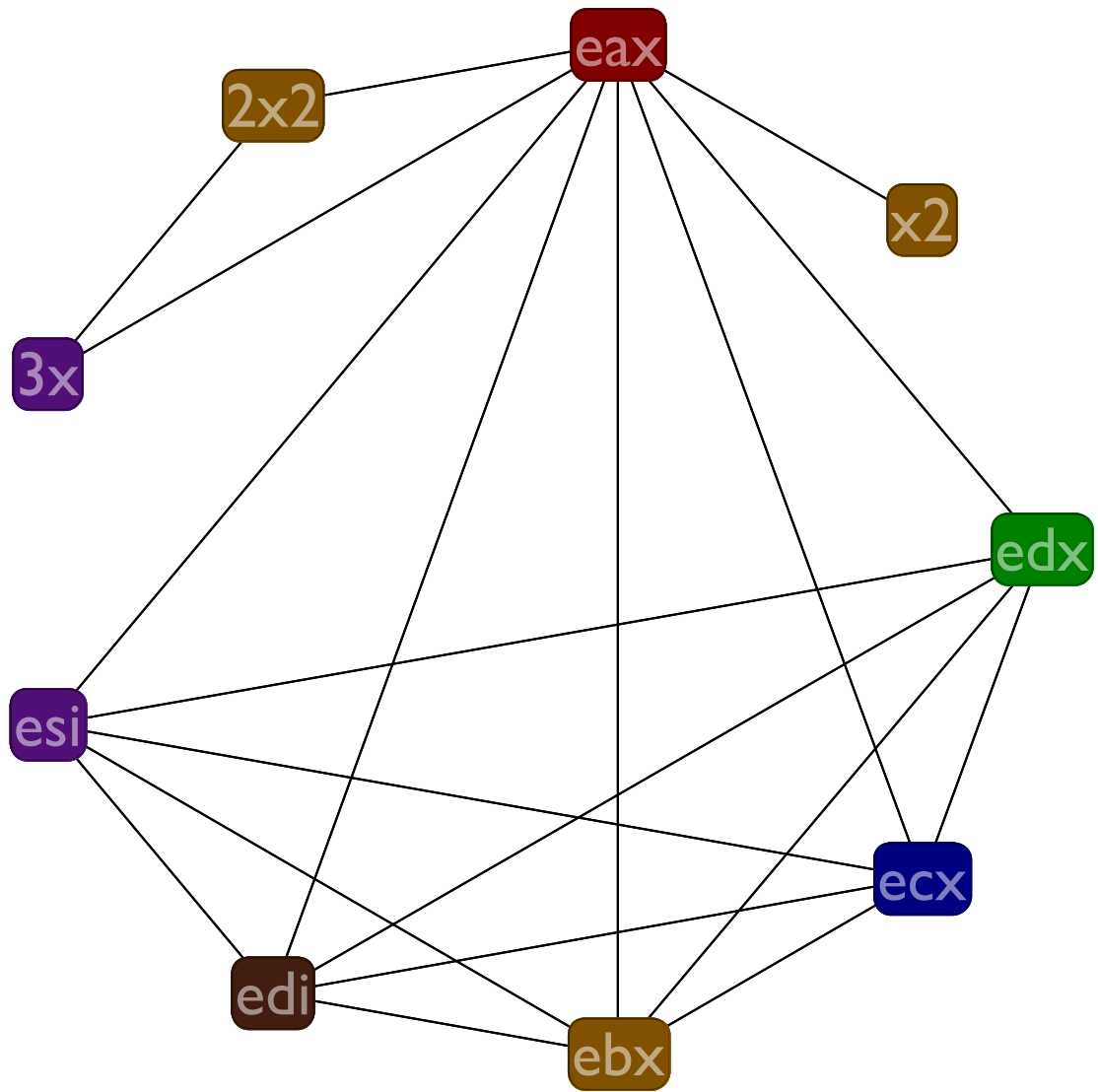












# Graph coloring algorithm

Some of the finer details of the graph coloring algorithm

- If possible, prefer to pull out nodes that have 5 or fewer edges at each stage
- When inserting & coloring nodes, always use the “smallest” color possible (according to any ordering on the colors you want). That is, imagine an ordering on the registers (say, alphabetical) and use the color of the first register in that order that works
- Ignore **ebp** and **esp** registers when building graph

# Caller and callee save registers

**Uhoh:** the result of coloring the example graph is wrong, since we didn't account for the callee save registers! For example, **3x** clobbers **esi**, breaking the calling convention.

# Gen/Kill overview

**caller save** ecx edx eax ebx

**args** eax edx ecx

**callee save** esi edi

**result** eax

	gen	kill
<b>(call s)</b>	<b>s</b> args	caller save result
<b>(tail-call s)</b>	<b>s</b> args callee save	
<b>(return)</b>	result callee save	

# Call may kill caller save regs

	gen	kill
<code>(call s)</code>	s args	caller save result
<code>(tail-call s)</code>	s args callee save	
<code>(return)</code>	result callee save	

```
(a <- 2)
(eax <- 1)
(call :f)
(a *= a)
```

Since the caller is responsible for saving the caller save registers, e.g., `ecx` then `:f` is free to clobber it. Thus we must be sure not to use `ecx` for `a`.

# Return gens result

	gen	kill
<code>(call s)</code>	s args	caller save result
<code>(tail-call s)</code>	s args callee save	
<code>(return)</code>	result callee save	

```
(eax <- 1)
(a <- edx)
(a *= a)
((mem b 4) <- a)
(return)
```

Must make sure that **a** is not allocated into the result register **eax**. With a reference (aka gen) at the **return** then the live range spans the assignment to **a** and thus **a** won't go into **eax**

# Ensure callee saves are saved

	gen	kill
<code>(call s)</code>	s args	caller save result
<code>(tail-call s)</code>	s args callee save	
<code>(return)</code>	result callee save	

```
(eax <- 1)
(a <- edx)
(a *= a)
((mem b 4) <- a)
(return)
```

Enforces the calling convention; specifically that the callee save registers are actually saved. In the code on the left, **a** must not be in **esi**



# Call kills return

	gen	kill
<code>(call s)</code>	s args	caller save result
<code>(tail-call s)</code>	s args callee save	
<code>(return)</code>	result callee save	

```
(a <- 1)  
(call :f)  
(b <- a)
```

If we put `a` into `eax` then the call to `:f` would clobber it to save the return value, so we must avoid that by treating the `call` as a kill to the return register, `eax`  
(Yep, this is the second reason to do this; other calling conventions may not have both reasons tho)

# Call/tail-call gen arguments

	gen	kill
<code>(call s)</code>	s args	caller save result
<code>(tail-call s)</code>	s args callee save	
<code>(return)</code>	result callee save	

```
(ecx <- 1)
```

```
(a <- 2)
```

```
(call :f)
```

If we put `a` into `ecx` then the call to `:f` would get the wrong arguments. Avoid this by making `call` and `tail-call` refer to (i.e., gen) the argument registers.

(Probably your compiler won't generate code like this, but spilling could put you into this kind of situation.)

# Constrained arithmetic operators

The `(cx <- s cmp s)` instruction in LI is limited to only 4 possible destinations.

The `(x shl= sh)` instruction in LI is limited to only shifting by the value of `ecx` (or by a constant in the other form).

# Constrained arithmetic operators

Add interference edges to disallow the illegal registers when building the interference graph, before starting the coloring.

E.g., if you have this instruction `(a <- y < x)` then add edges between `a` and the registers `edi` and `esi`, ensuring `a` ends up in `eax`, `ecx`, `edx`, `ebx`, or spilled.

# Do over

Lets redo the coloring, now with the callee and caller  
save register information in the graph

# Gen & Kill

	<b>gen</b>	<b>kill</b>
1: <code>:f</code>	<code>()</code>	<code>()</code>
2: <code>(x2 &lt;- eax)</code>	<code>(eax)</code>	<code>(x2)</code>
3: <code>(x2 *= x2)</code>	<code>(x2)</code>	<code>(x2)</code>
4: <code>(2x2 &lt;- x2)</code>	<code>(x2)</code>	<code>(2x2)</code>
5: <code>(2x2 *= 2)</code>	<code>(2x2)</code>	<code>(2x2)</code>
6: <code>(3x &lt;- eax)</code>	<code>(eax)</code>	<code>(3x)</code>
7: <code>(3x *= 3)</code>	<code>(3x)</code>	<code>(3x)</code>
8: <code>(eax &lt;- 2x2)</code>	<code>(2x2)</code>	<code>(eax)</code>
9: <code>(eax += 3x)</code>	<code>(3x eax)</code>	<code>(eax)</code>
10: <code>(eax += 4)</code>	<code>(eax)</code>	<code>(eax)</code>
11: <code>(return)</code>	<code>(eax edi esi)</code>	<code>()</code>

# Liveness

	<b>in</b>	<b>out</b>
1: :f	()	()
2: (x2 <- eax)	()	()
3: (x2 *= x2)	()	()
4: (2x2 <- x2)	()	()
5: (2x2 *= 2)	()	()
6: (3x <- eax)	()	()
7: (3x *= 3)	()	()
8: (eax <- 2x2)	()	()
9: (eax += 3x)	()	()
10: (eax += 4)	()	()
11: (return)	()	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	()	()
2: <code>(x2 &lt;- eax)</code>	(eax)	()
3: <code>(x2 *= x2)</code>	(x2)	()
4: <code>(2x2 &lt;- x2)</code>	(x2)	()
5: <code>(2x2 *= 2)</code>	(2x2)	()
6: <code>(3x &lt;- eax)</code>	(eax)	()
7: <code>(3x *= 3)</code>	(3x)	()
8: <code>(eax &lt;- 2x2)</code>	(2x2)	()
9: <code>(eax += 3x)</code>	(3x eax)	()
10: <code>(eax += 4)</code>	(eax)	()
11: <code>(return)</code>	(eax edi esi)	()



# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	<code>()</code>	<code>(eax)</code>
2: <code>(x2 &lt;- eax)</code>	<code>(eax)</code>	<code>(x2)</code>
3: <code>(x2 *= x2)</code>	<code>(x2)</code>	<code>(x2)</code>
4: <code>(2x2 &lt;- x2)</code>	<code>(x2)</code>	<code>(2x2)</code>
5: <code>(2x2 *= 2)</code>	<code>(2x2)</code>	<code>(eax)</code>
6: <code>(3x &lt;- eax)</code>	<code>(eax)</code>	<code>(3x)</code>
7: <code>(3x *= 3)</code>	<code>(3x)</code>	<code>(2x2)</code>
8: <code>(eax &lt;- 2x2)</code>	<code>(2x2)</code>	<code>(3x eax)</code>
9: <code>(eax += 3x)</code>	<code>(3x eax)</code>	<code>(eax)</code>
10: <code>(eax += 4)</code>	<code>(eax)</code>	<code>(eax edi esi)</code>
11: <code>(return)</code>	<code>(eax edi esi)</code>	<code>()</code>

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(x2)
3: <code>(x2 *= x2)</code>	(x2)	(x2)
4: <code>(2x2 &lt;- x2)</code>	(x2)	(2x2)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(eax)
6: <code>(3x &lt;- eax)</code>	(eax)	(3x)
7: <code>(3x *= 3)</code>	(2x2 3x)	(2x2)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x)	(3x eax)
9: <code>(eax += 3x)</code>	(3x eax)	(eax)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(x2)
3: <code>(x2 *= x2)</code>	(x2)	(x2)
4: <code>(2x2 &lt;- x2)</code>	(x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(eax)
6: <code>(3x &lt;- eax)</code>	(eax)	(2x2 3x)
7: <code>(3x *= 3)</code>	(2x2 3x)	(2x2 3x)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x)	(3x eax)
9: <code>(eax += 3x)</code>	(3x eax)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(x2)
3: <code>(x2 *= x2)</code>	(x2)	(x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(eax)
6: <code>(3x &lt;- eax)</code>	(2x2 eax)	(2x2 3x)
7: <code>(3x *= 3)</code>	(2x2 3x)	(2x2 3x)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x)	(3x eax)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(x2)
3: <code>(x2 *= x2)</code>	(x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(2x2 eax)
6: <code>(3x &lt;- eax)</code>	(2x2 eax)	(2x2 3x)
7: <code>(3x *= 3)</code>	(2x2 3x)	(2x2 3x)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(x2)
3: <code>(x2 *= x2)</code>	(eax x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(2x2 eax)
6: <code>(3x &lt;- eax)</code>	(2x2 eax)	(2x2 3x)
7: <code>(3x *= 3)</code>	(2x2 3x)	(2x2 3x)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(eax x2)
3: <code>(x2 *= x2)</code>	(eax x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(2x2 eax)
6: <code>(3x &lt;- eax)</code>	(2x2 eax)	(2x2 3x)
7: <code>(3x *= 3)</code>	(2x2 3x)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(eax x2)
3: <code>(x2 *= x2)</code>	(eax x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(2x2 eax)
6: <code>(3x &lt;- eax)</code>	(2x2 eax)	(2x2 3x)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()



# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(eax x2)
3: <code>(x2 *= x2)</code>	(eax x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(2x2 eax)
6: <code>(3x &lt;- eax)</code>	(2x2 eax)	(2x2 3x edi esi)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(eax x2)
3: <code>(x2 *= x2)</code>	(eax x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(2x2 eax)
6: <code>(3x &lt;- eax)</code>	(2x2 eax edi esi)	(2x2 3x edi esi)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(eax x2)
3: <code>(x2 *= x2)</code>	(eax x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax)	(2x2 eax edi esi)
6: <code>(3x &lt;- eax)</code>	(2x2 eax edi esi)	(2x2 3x edi esi)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(eax x2)
3: <code>(x2 *= x2)</code>	(eax x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax)
5: <code>(2x2 *= 2)</code>	(2x2 eax edi esi)	(2x2 eax edi esi)
6: <code>(3x &lt;- eax)</code>	(2x2 eax edi esi)	(2x2 3x edi esi)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(eax x2)
3: <code>(x2 *= x2)</code>	(eax x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax x2)	(2x2 eax edi esi)
5: <code>(2x2 *= 2)</code>	(2x2 eax edi esi)	(2x2 eax edi esi)
6: <code>(3x &lt;- eax)</code>	(2x2 eax edi esi)	(2x2 3x edi esi)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(eax x2)
3: <code>(x2 *= x2)</code>	(eax x2)	(eax x2)
4: <code>(2x2 &lt;- x2)</code>	(eax edi esi x2)	(2x2 eax edi esi)
5: <code>(2x2 *= 2)</code>	(2x2 eax edi esi)	(2x2 eax edi esi)
6: <code>(3x &lt;- eax)</code>	(2x2 eax edi esi)	(2x2 3x edi esi)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(eax x2)
3: <code>(x2 *= x2)</code>	(eax x2)	(eax edi esi x2)
4: <code>(2x2 &lt;- x2)</code>	(eax edi esi x2)	(2x2 eax edi esi)
5: <code>(2x2 *= 2)</code>	(2x2 eax edi esi)	(2x2 eax edi esi)
6: <code>(3x &lt;- eax)</code>	(2x2 eax edi esi)	(2x2 3x edi esi)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(eax x2)
3: <code>(x2 *= x2)</code>	(eax edi esi x2)	(eax edi esi x2)
4: <code>(2x2 &lt;- x2)</code>	(eax edi esi x2)	(2x2 eax edi esi)
5: <code>(2x2 *= 2)</code>	(2x2 eax edi esi)	(2x2 eax edi esi)
6: <code>(3x &lt;- eax)</code>	(2x2 eax edi esi)	(2x2 3x edi esi)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()



# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax)	(eax edi esi x2)
3: <code>(x2 *= x2)</code>	(eax edi esi x2)	(eax edi esi x2)
4: <code>(2x2 &lt;- x2)</code>	(eax edi esi x2)	(2x2 eax edi esi)
5: <code>(2x2 *= 2)</code>	(2x2 eax edi esi)	(2x2 eax edi esi)
6: <code>(3x &lt;- eax)</code>	(2x2 eax edi esi)	(2x2 3x edi esi)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

# Liveness

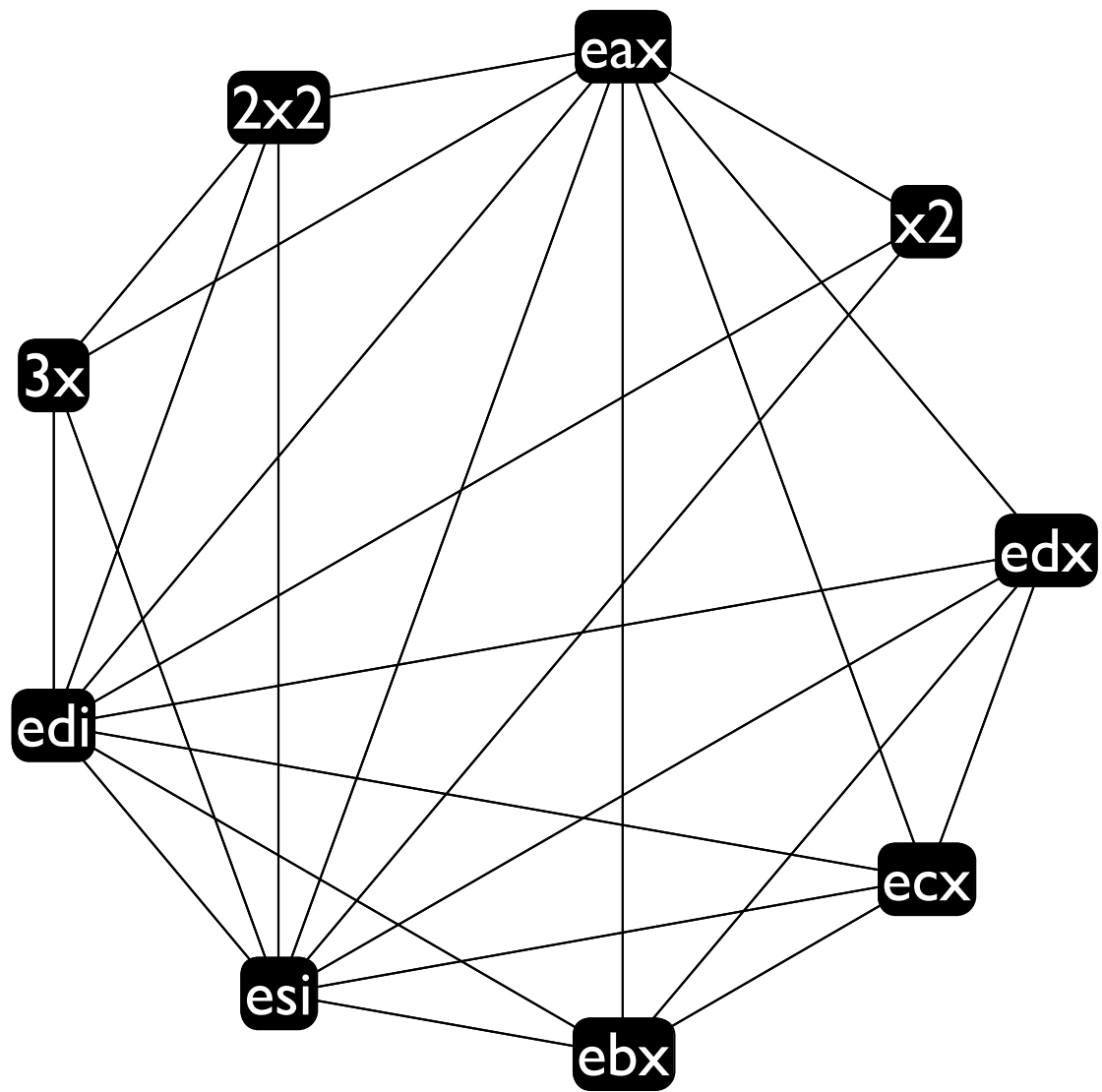
	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax)
2: <code>(x2 &lt;- eax)</code>	(eax edi esi)	(eax edi esi x2)
3: <code>(x2 *= x2)</code>	(eax edi esi x2)	(eax edi esi x2)
4: <code>(2x2 &lt;- x2)</code>	(eax edi esi x2)	(2x2 eax edi esi)
5: <code>(2x2 *= 2)</code>	(2x2 eax edi esi)	(2x2 eax edi esi)
6: <code>(3x &lt;- eax)</code>	(2x2 eax edi esi)	(2x2 3x edi esi)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

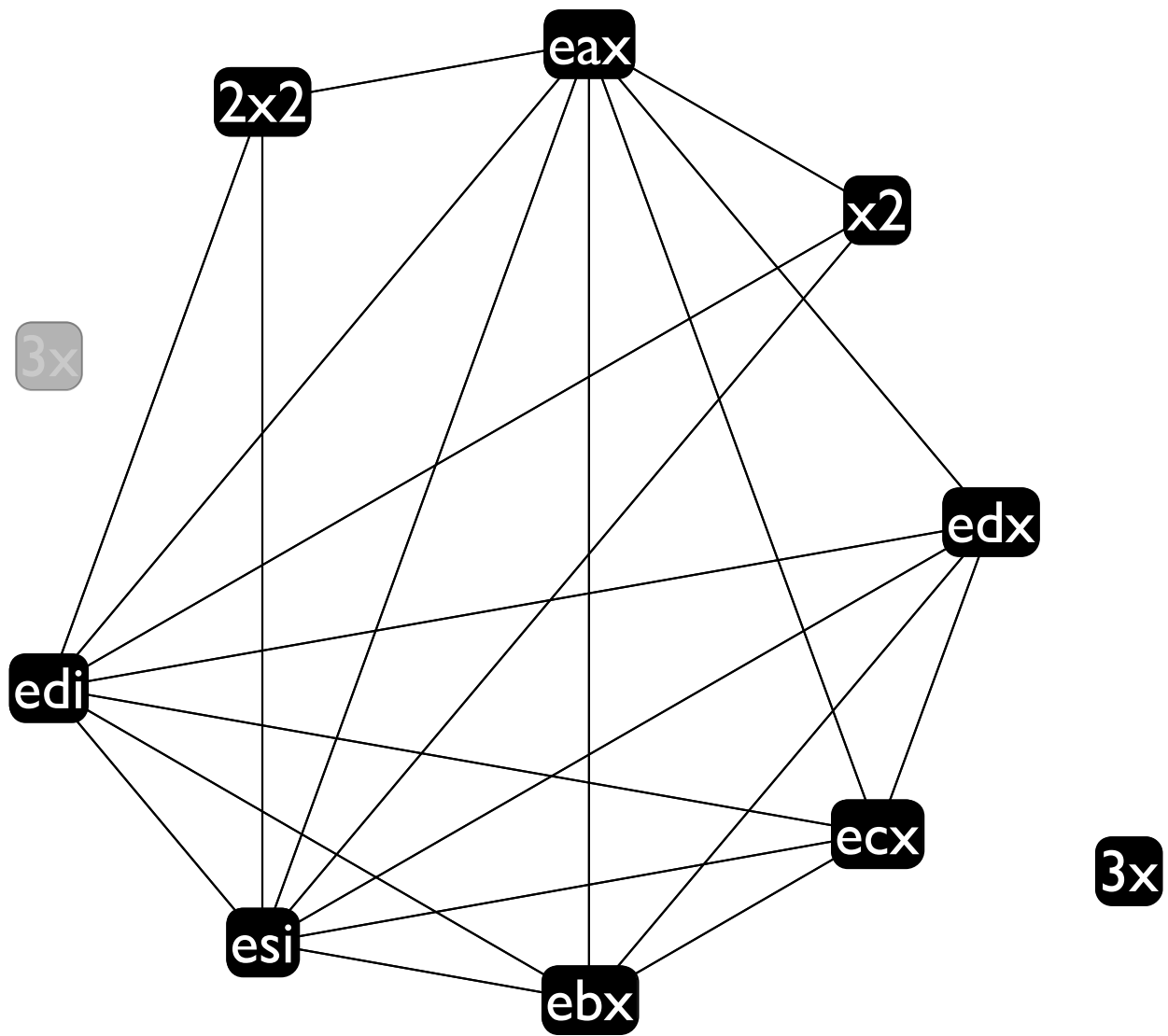
# Liveness

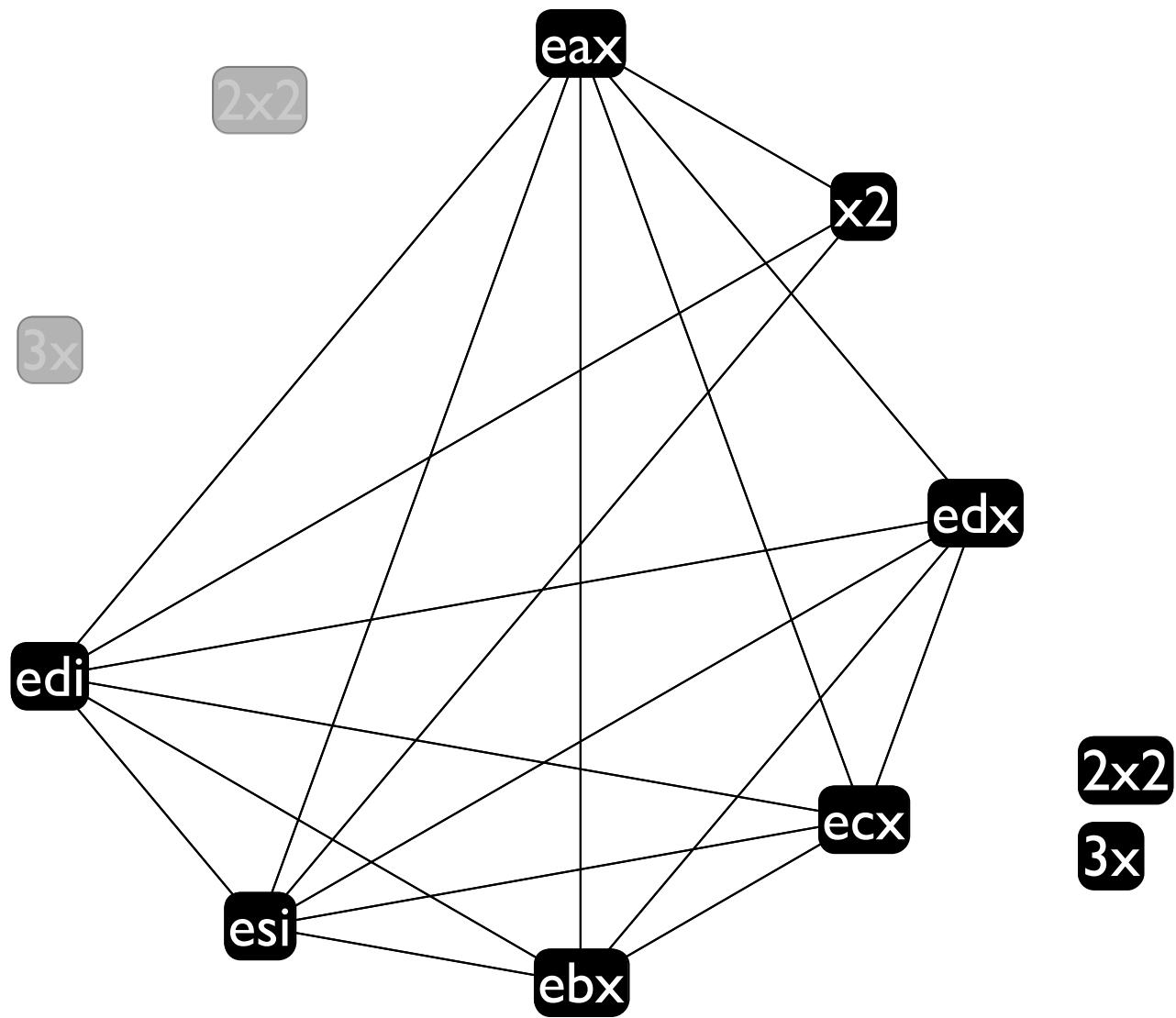
	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax)	(eax edi esi)
2: <code>(x2 &lt;- eax)</code>	(eax edi esi)	(eax edi esi x2)
3: <code>(x2 *= x2)</code>	(eax edi esi x2)	(eax edi esi x2)
4: <code>(2x2 &lt;- x2)</code>	(eax edi esi x2)	(2x2 eax edi esi)
5: <code>(2x2 *= 2)</code>	(2x2 eax edi esi)	(2x2 eax edi esi)
6: <code>(3x &lt;- eax)</code>	(2x2 eax edi esi)	(2x2 3x edi esi)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

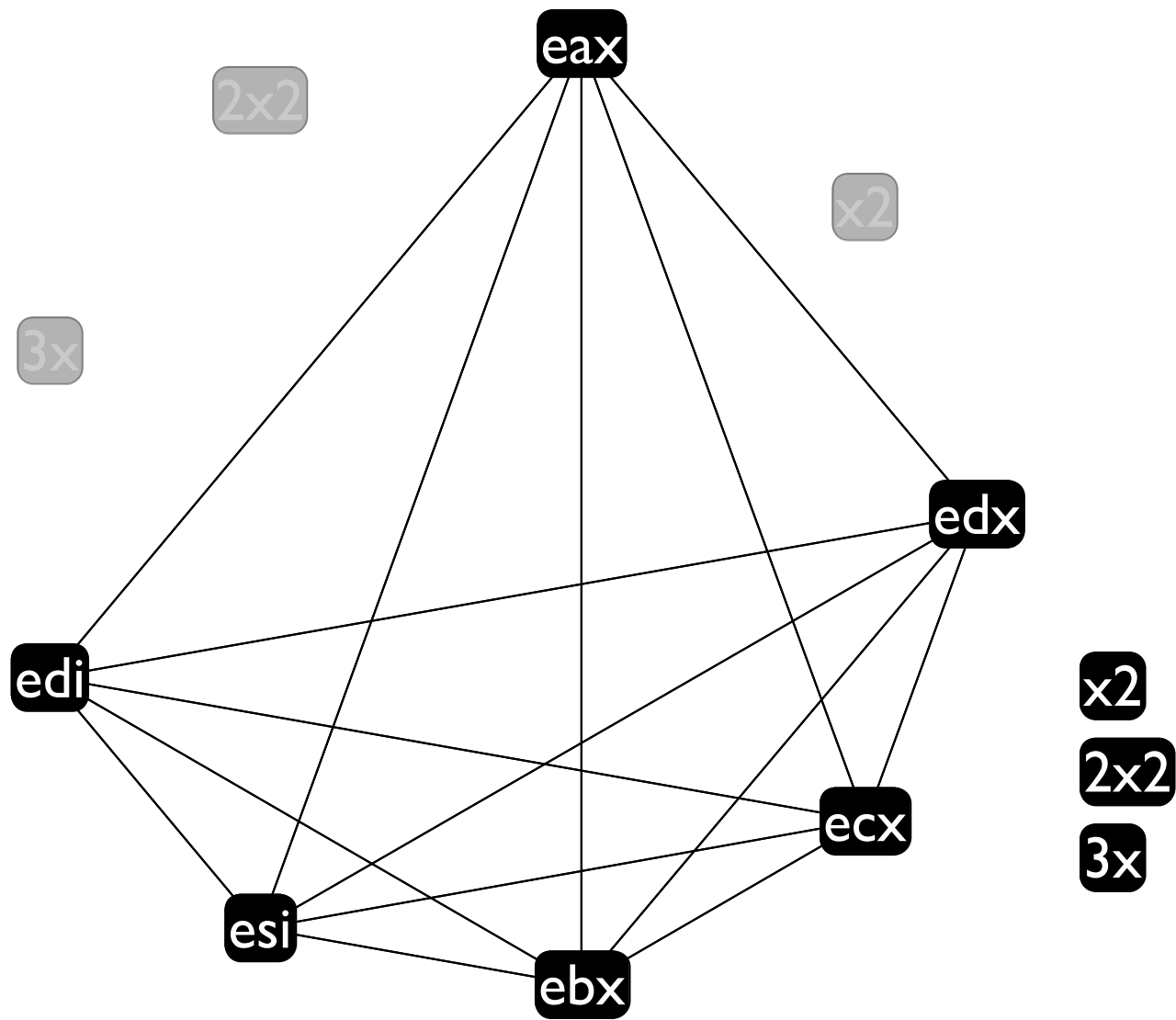
# Liveness

	<b>in</b>	<b>out</b>
1: <code>:f</code>	(eax edi esi)	(eax edi esi)
2: <code>(x2 &lt;- eax)</code>	(eax edi esi)	(eax edi esi x2)
3: <code>(x2 *= x2)</code>	(eax edi esi x2)	(eax edi esi x2)
4: <code>(2x2 &lt;- x2)</code>	(eax edi esi x2)	(2x2 eax edi esi)
5: <code>(2x2 *= 2)</code>	(2x2 eax edi esi)	(2x2 eax edi esi)
6: <code>(3x &lt;- eax)</code>	(2x2 eax edi esi)	(2x2 3x edi esi)
7: <code>(3x *= 3)</code>	(2x2 3x edi esi)	(2x2 3x edi esi)
8: <code>(eax &lt;- 2x2)</code>	(2x2 3x edi esi)	(3x eax edi esi)
9: <code>(eax += 3x)</code>	(3x eax edi esi)	(eax edi esi)
10: <code>(eax += 4)</code>	(eax edi esi)	(eax edi esi)
11: <code>(return)</code>	(eax edi esi)	()

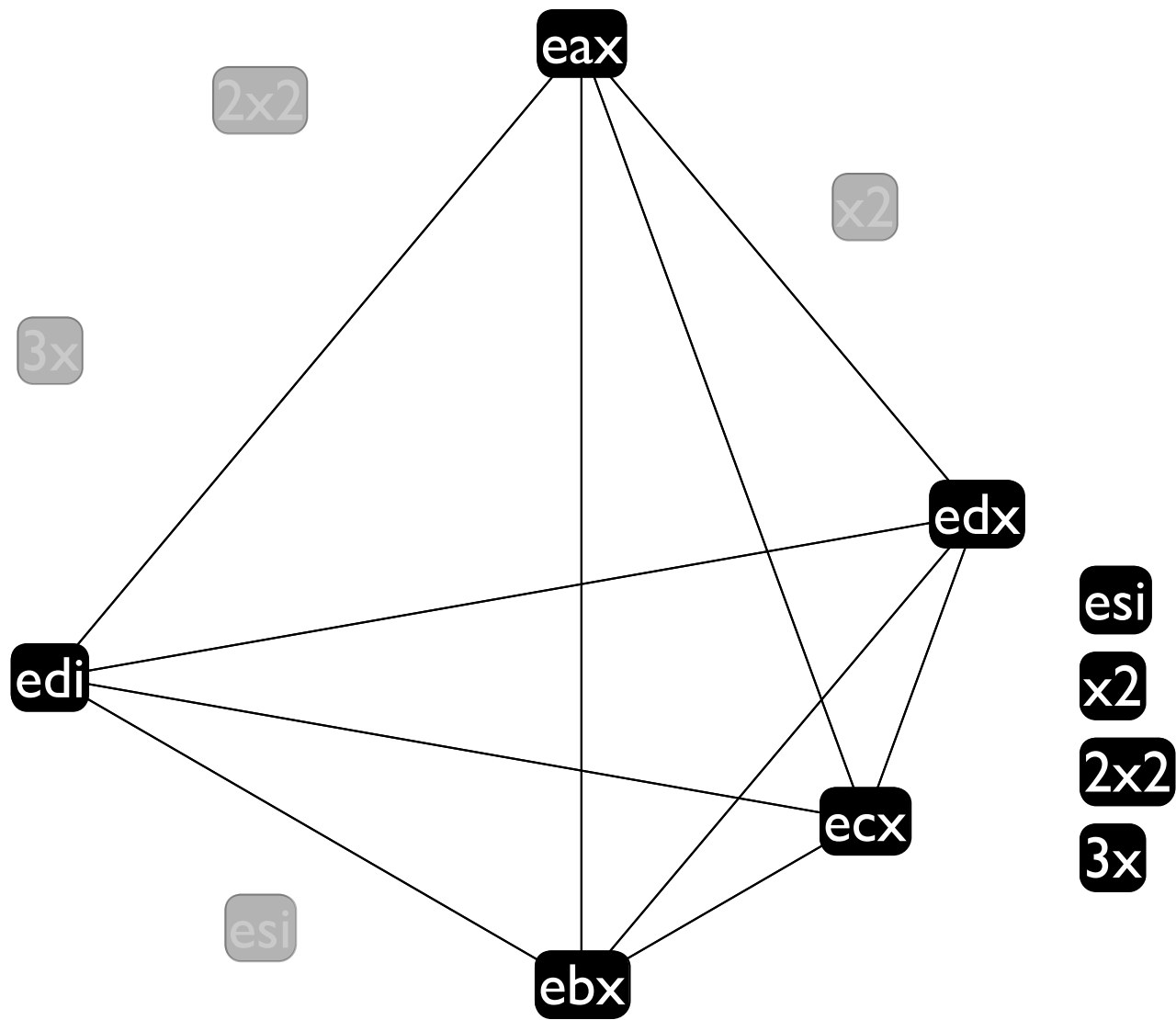


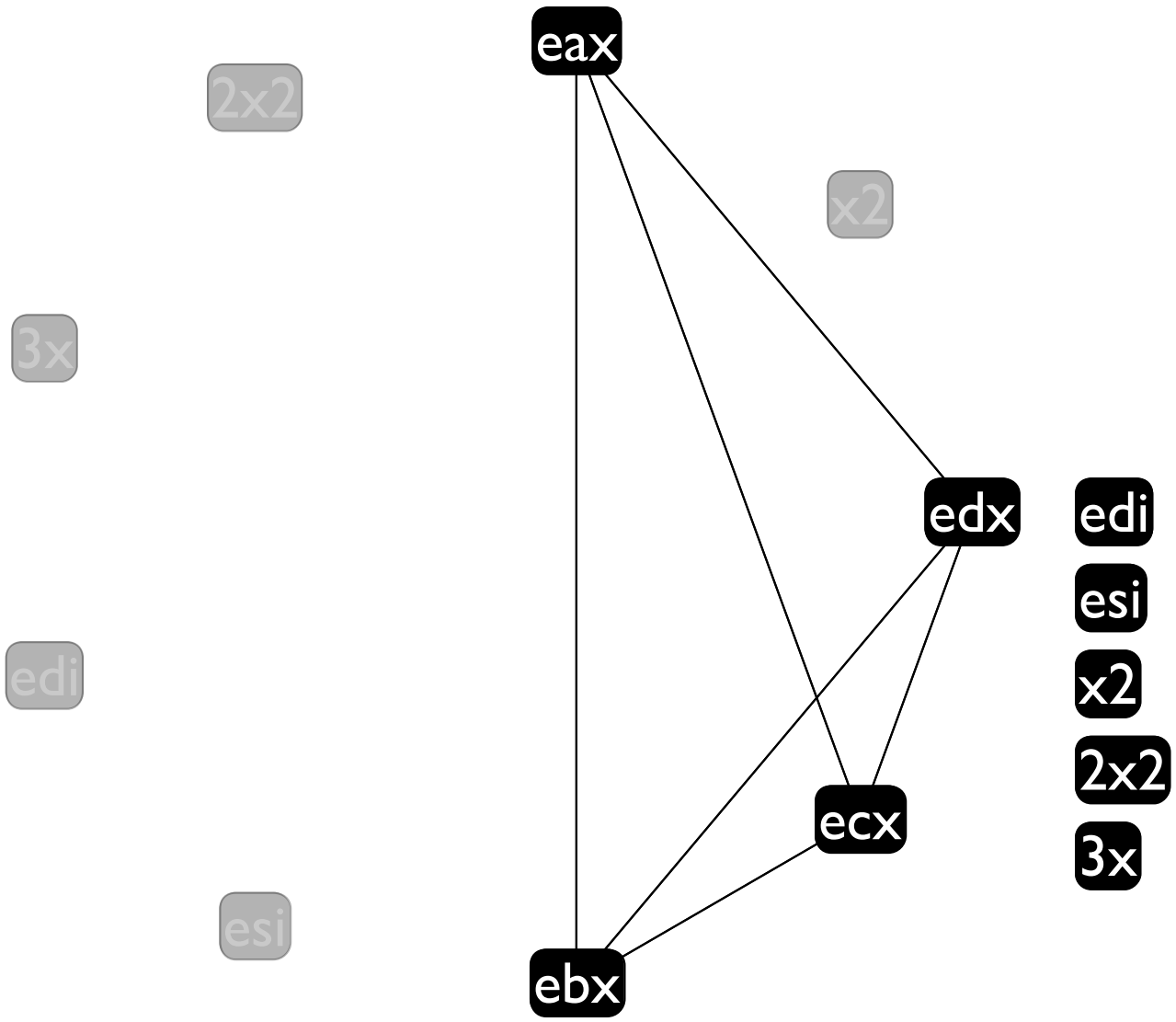


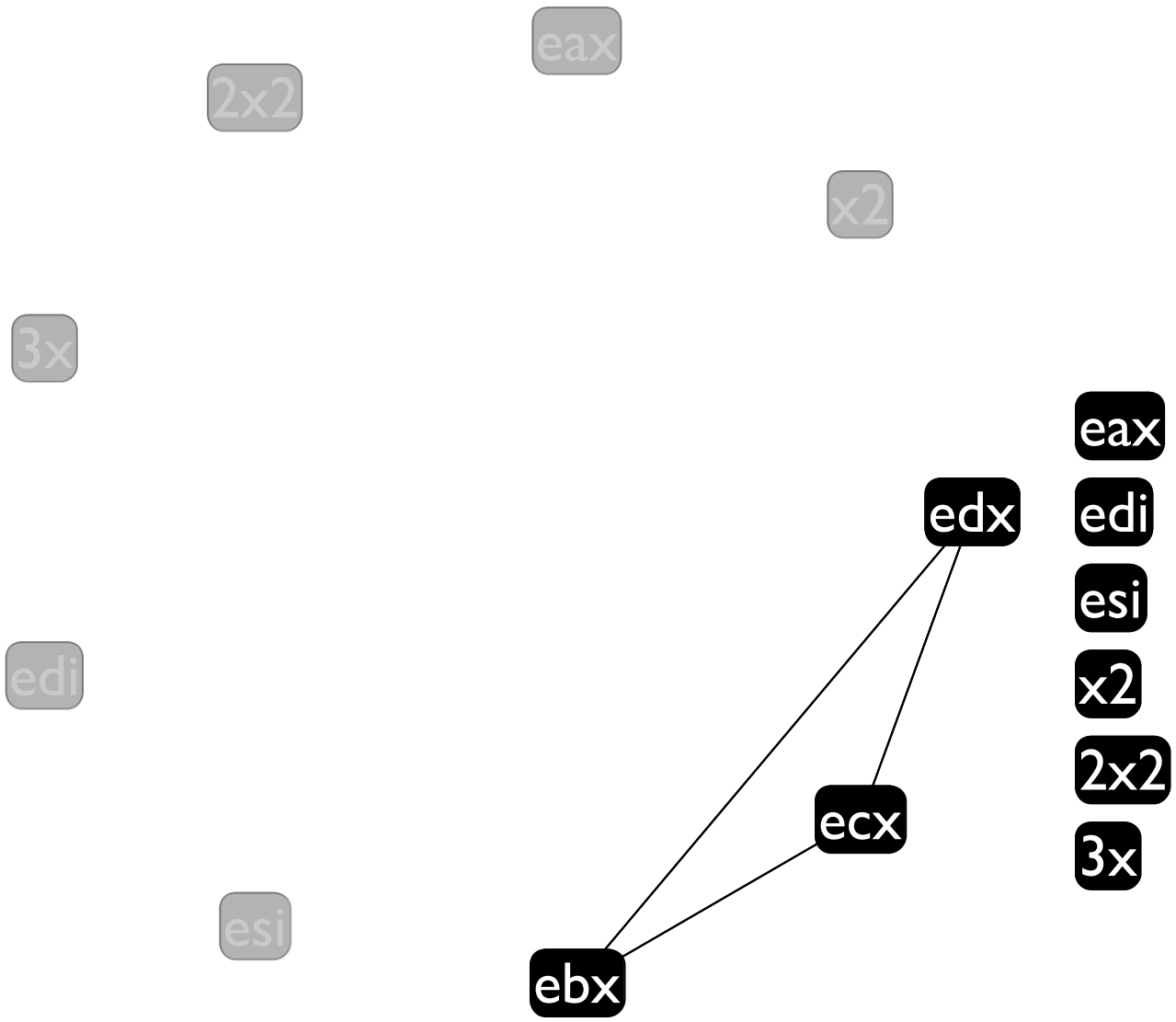


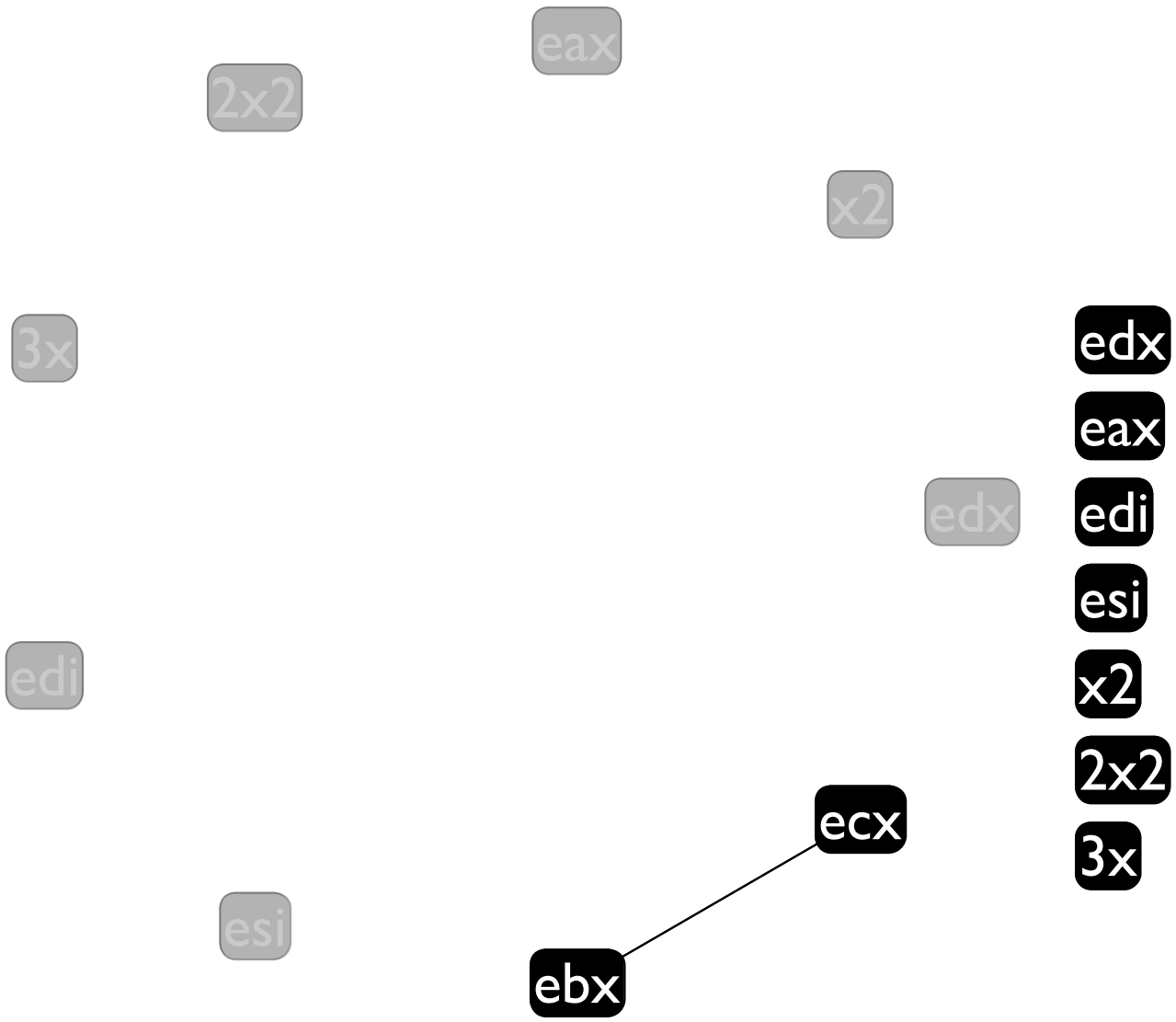


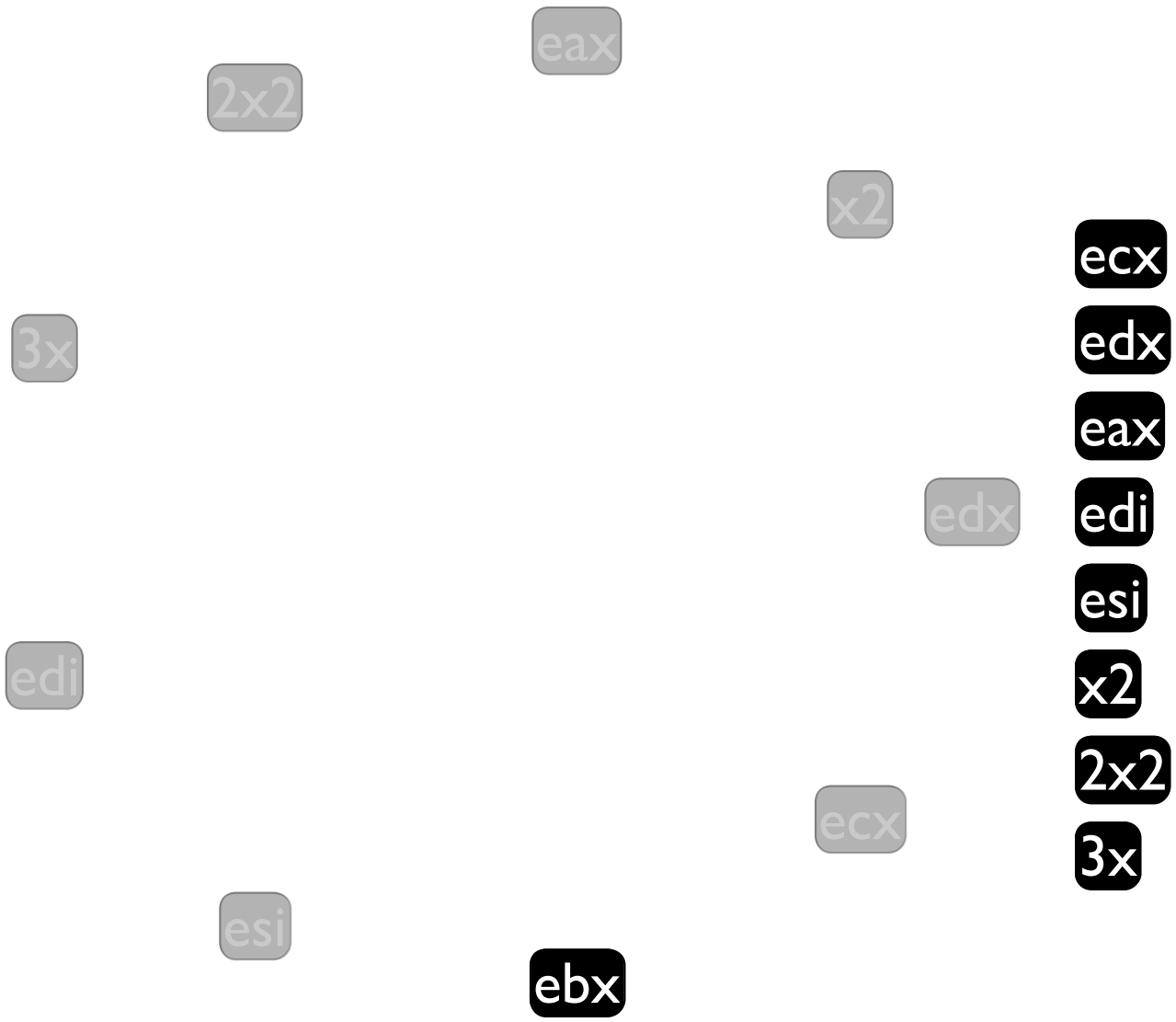


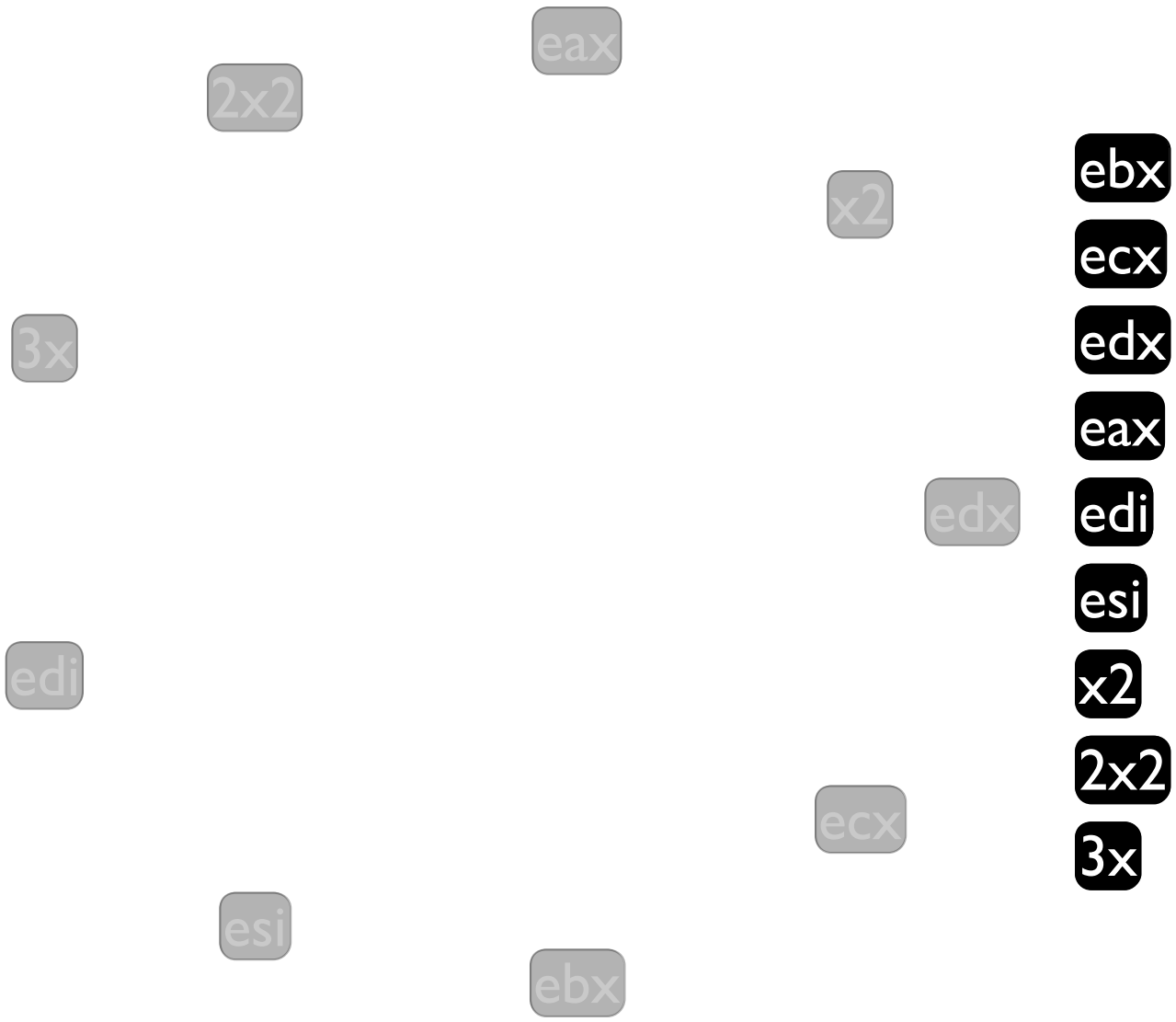


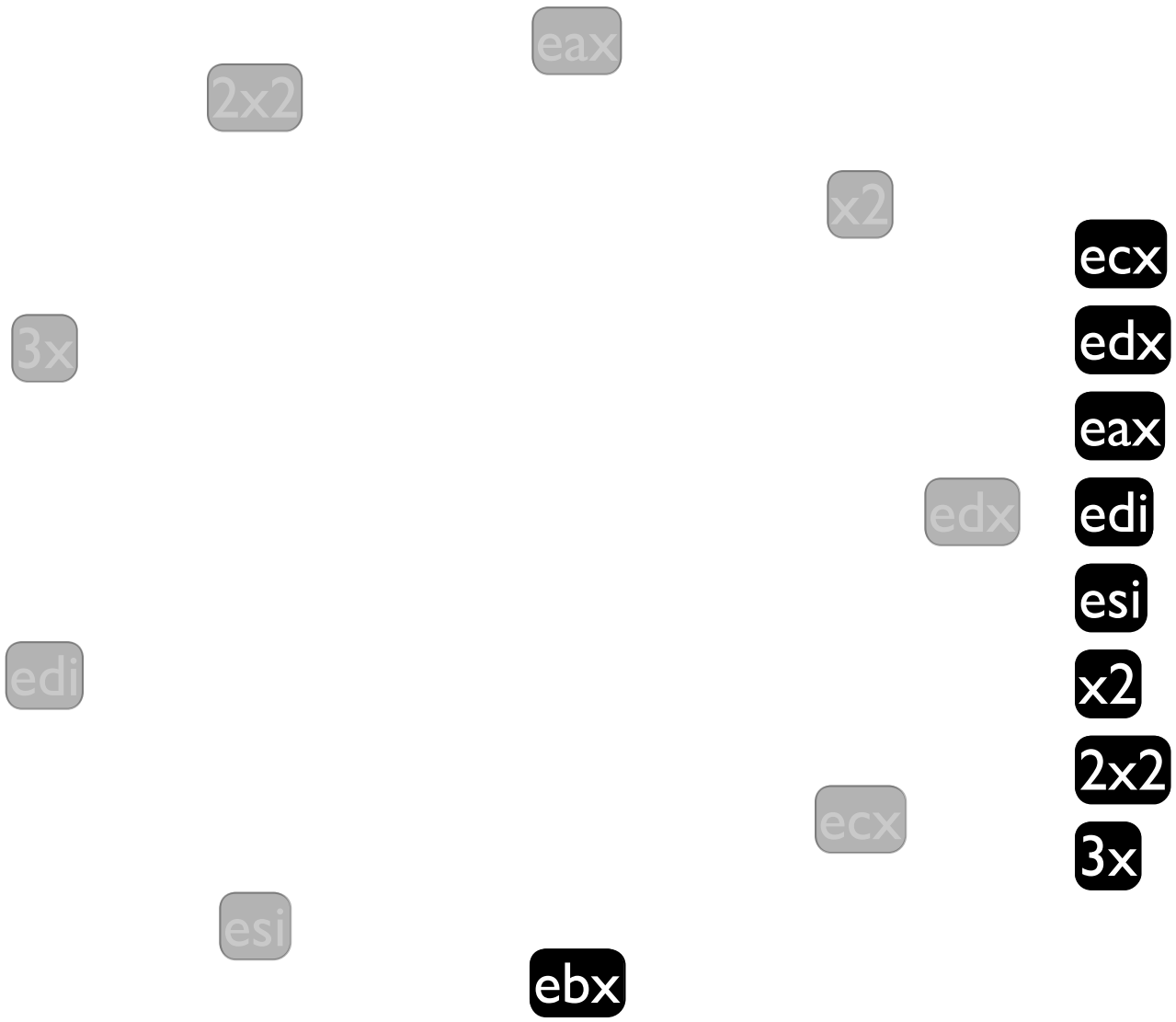


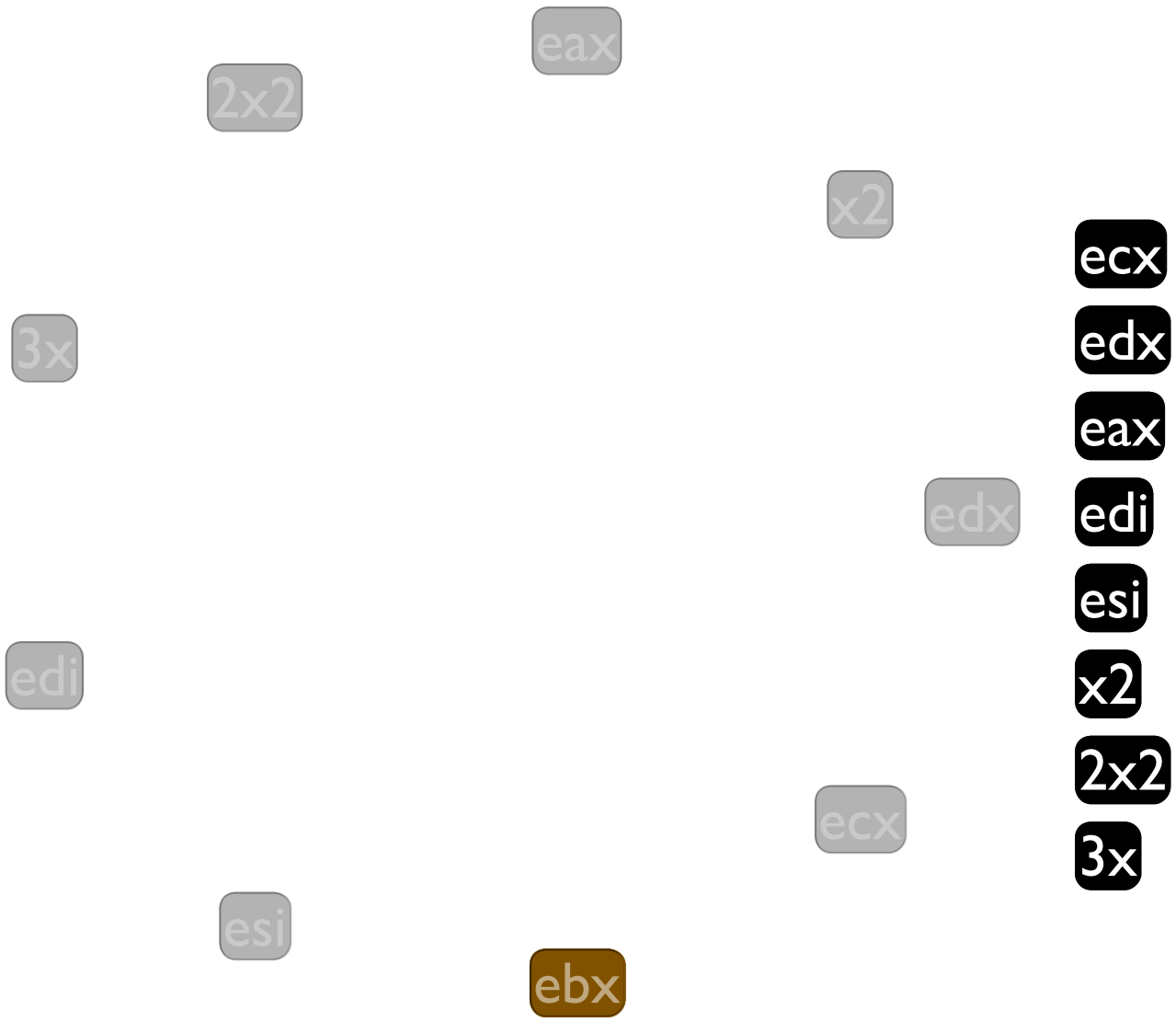




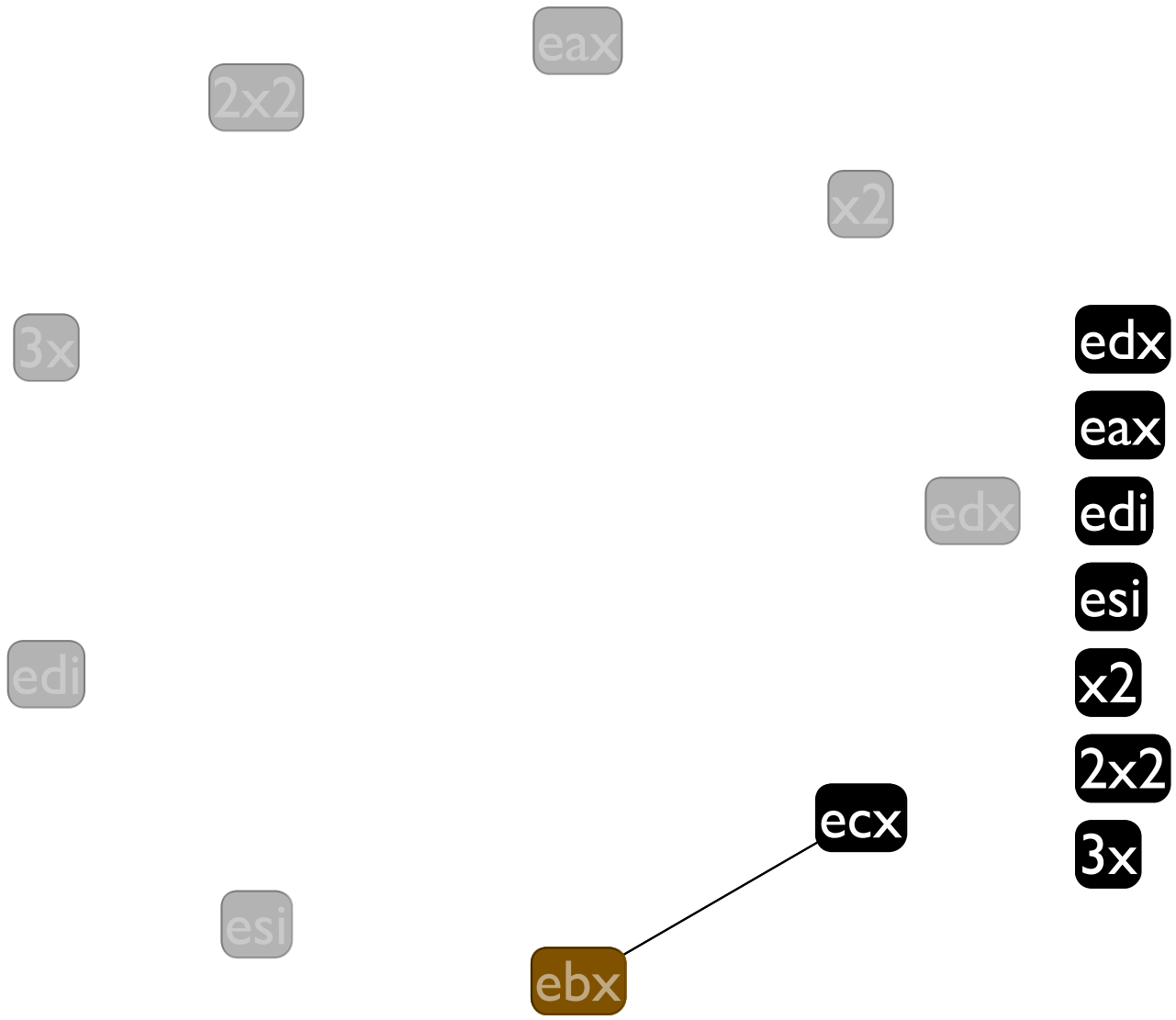


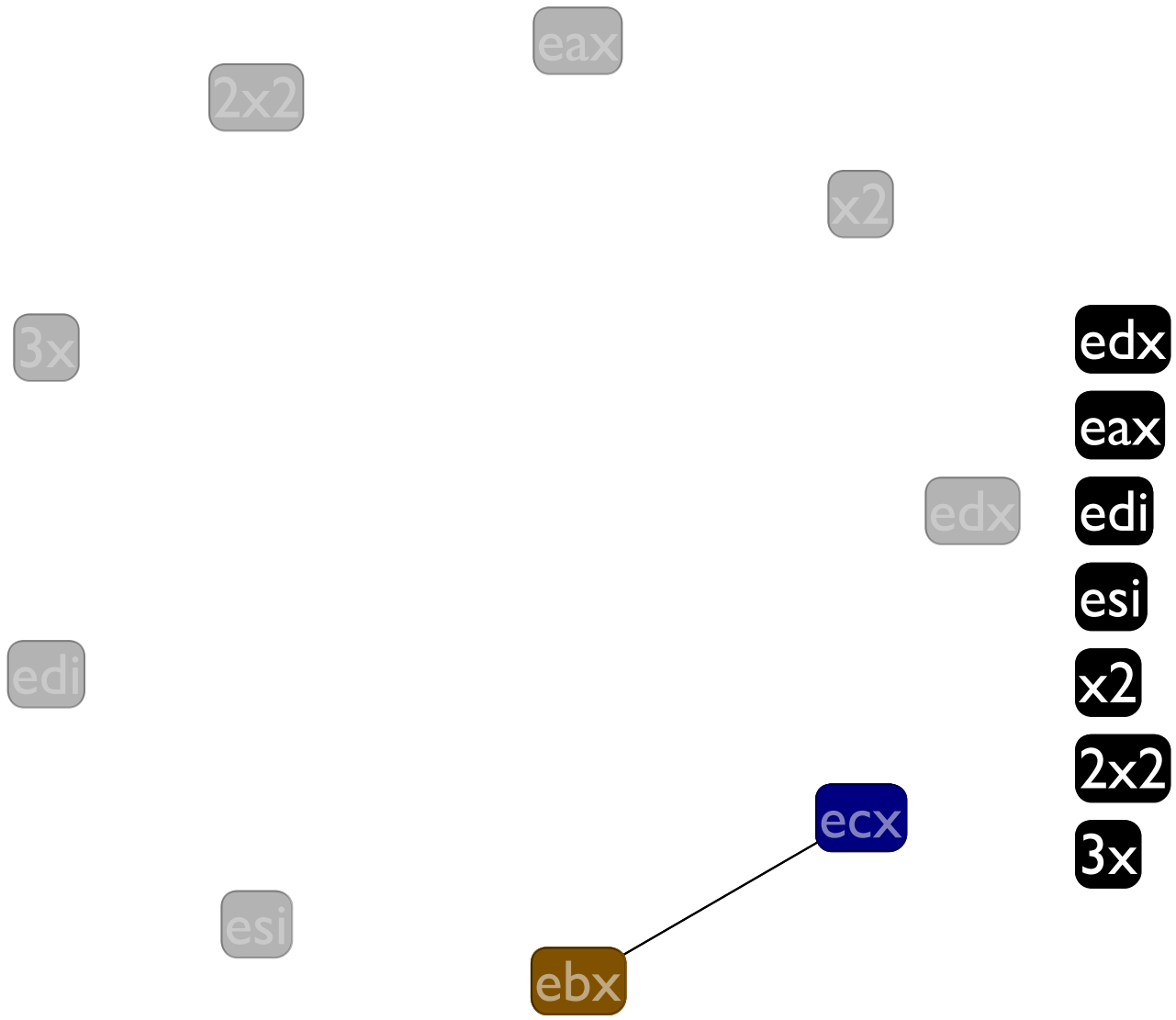


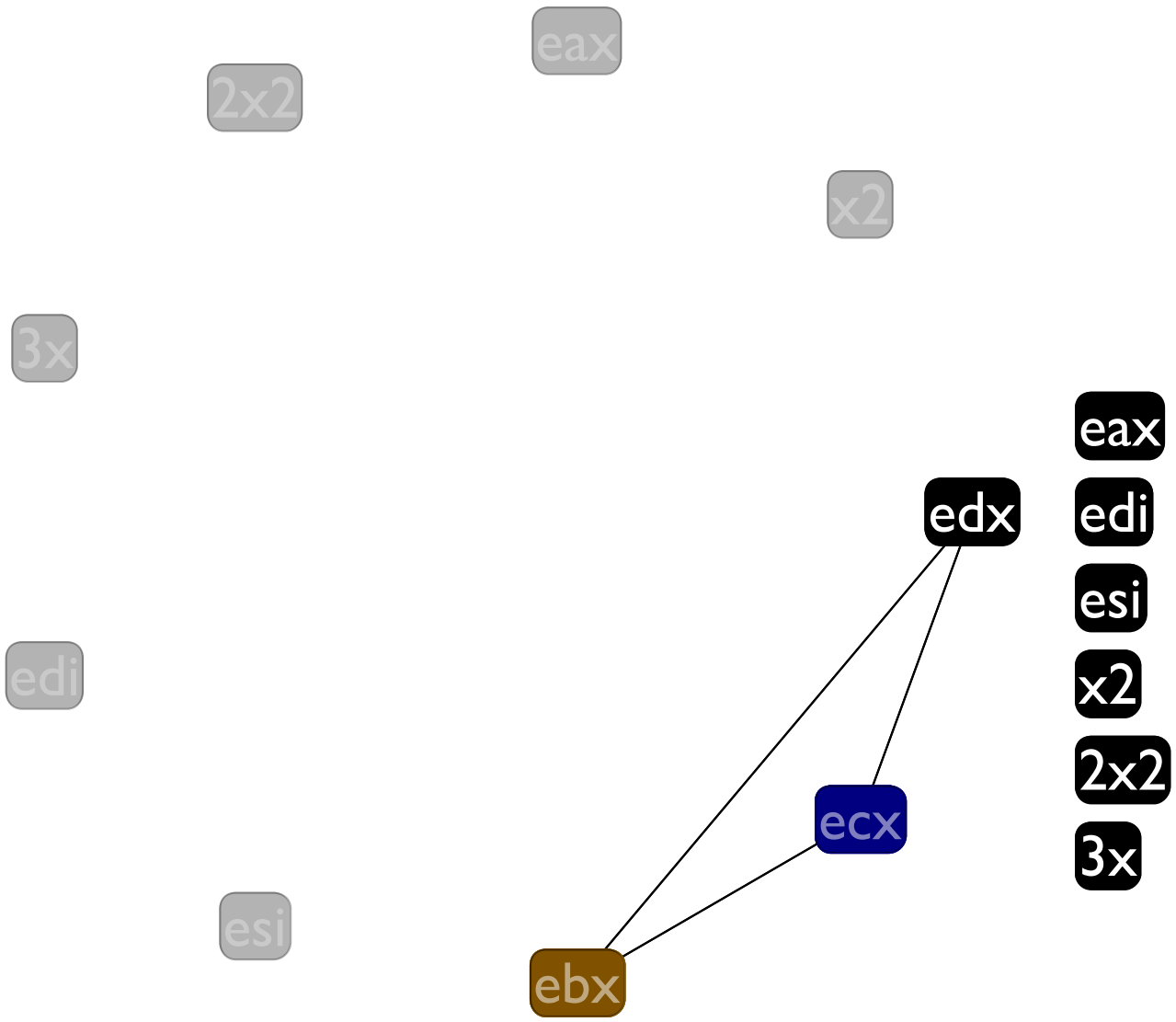


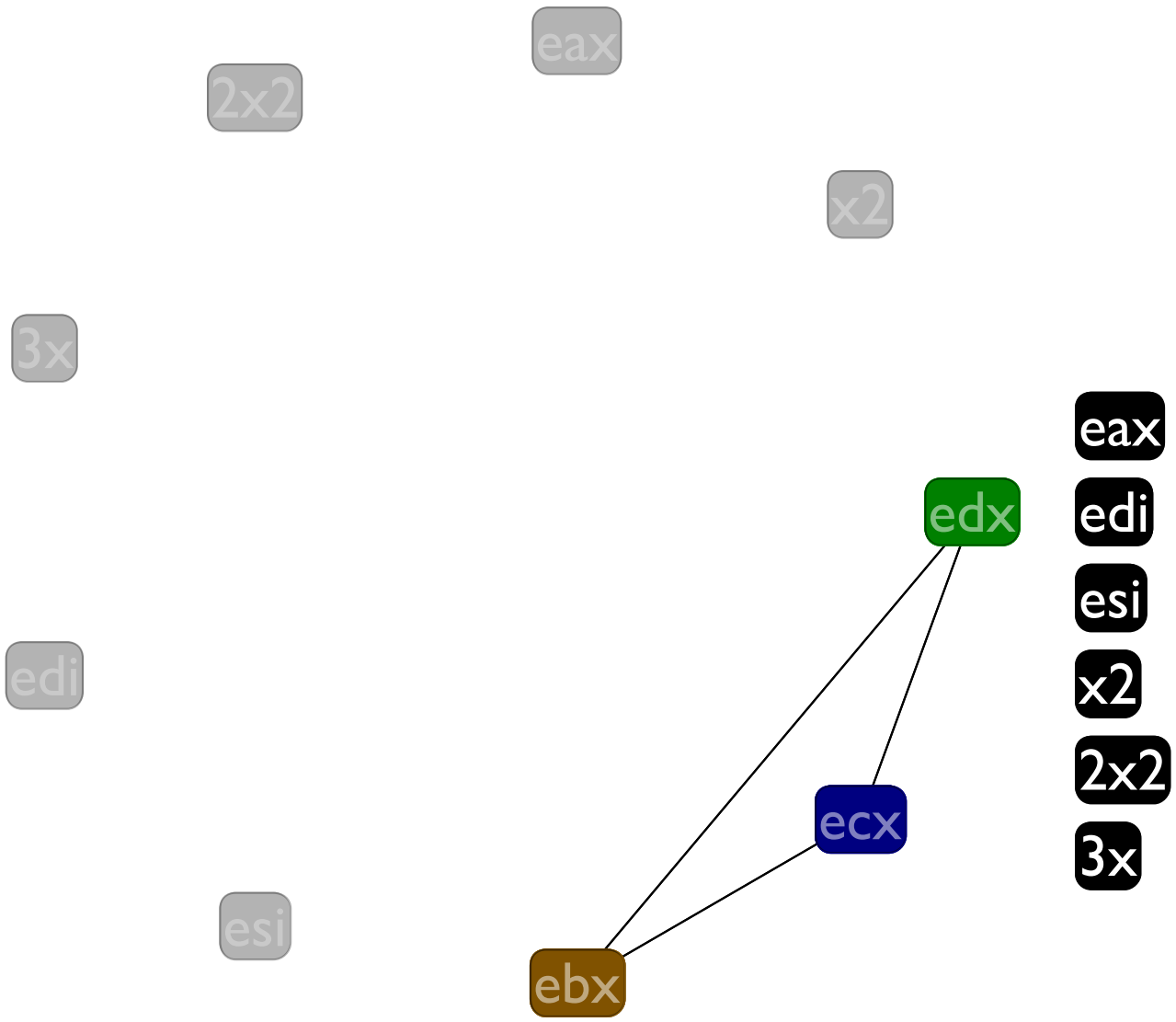


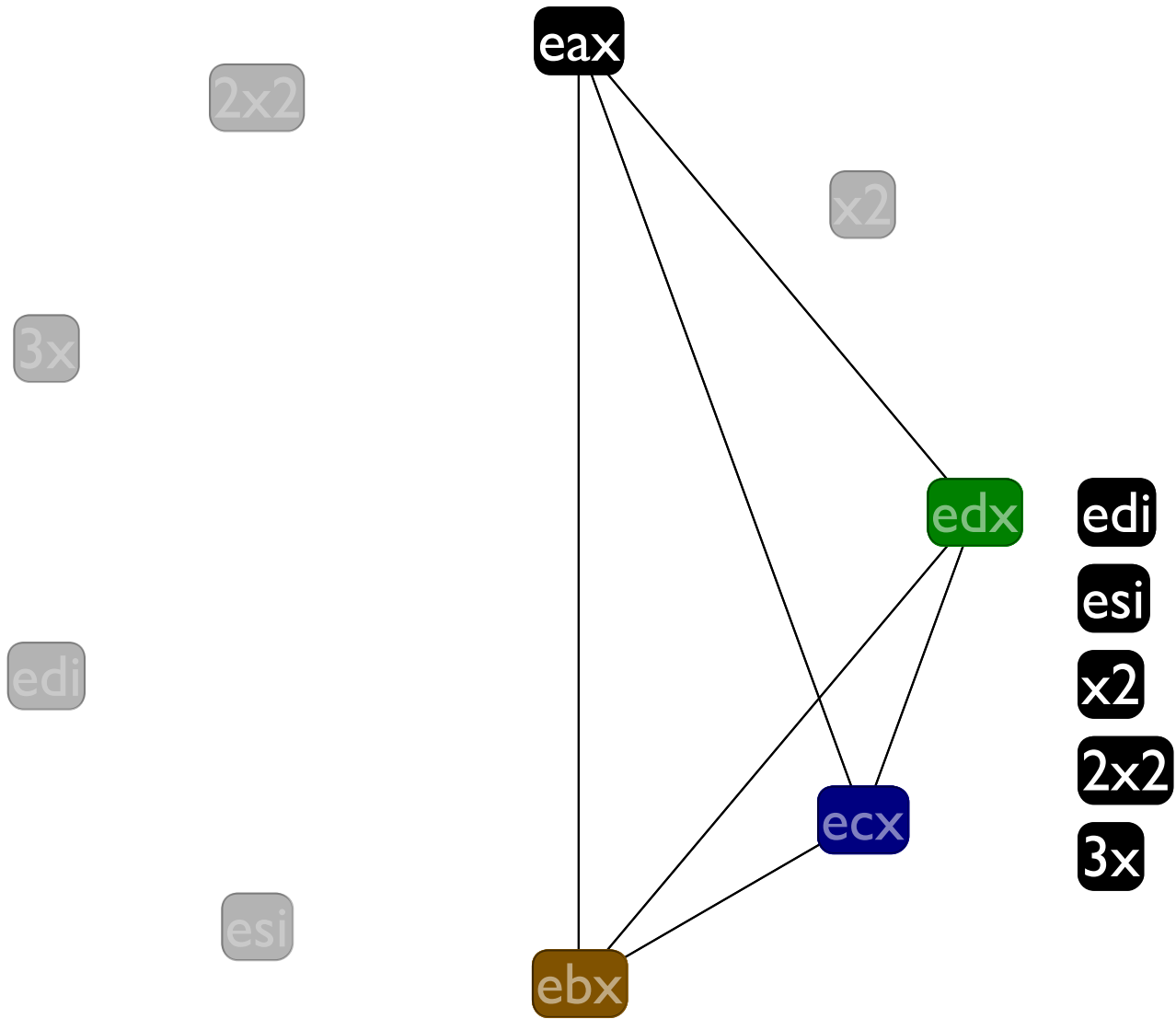


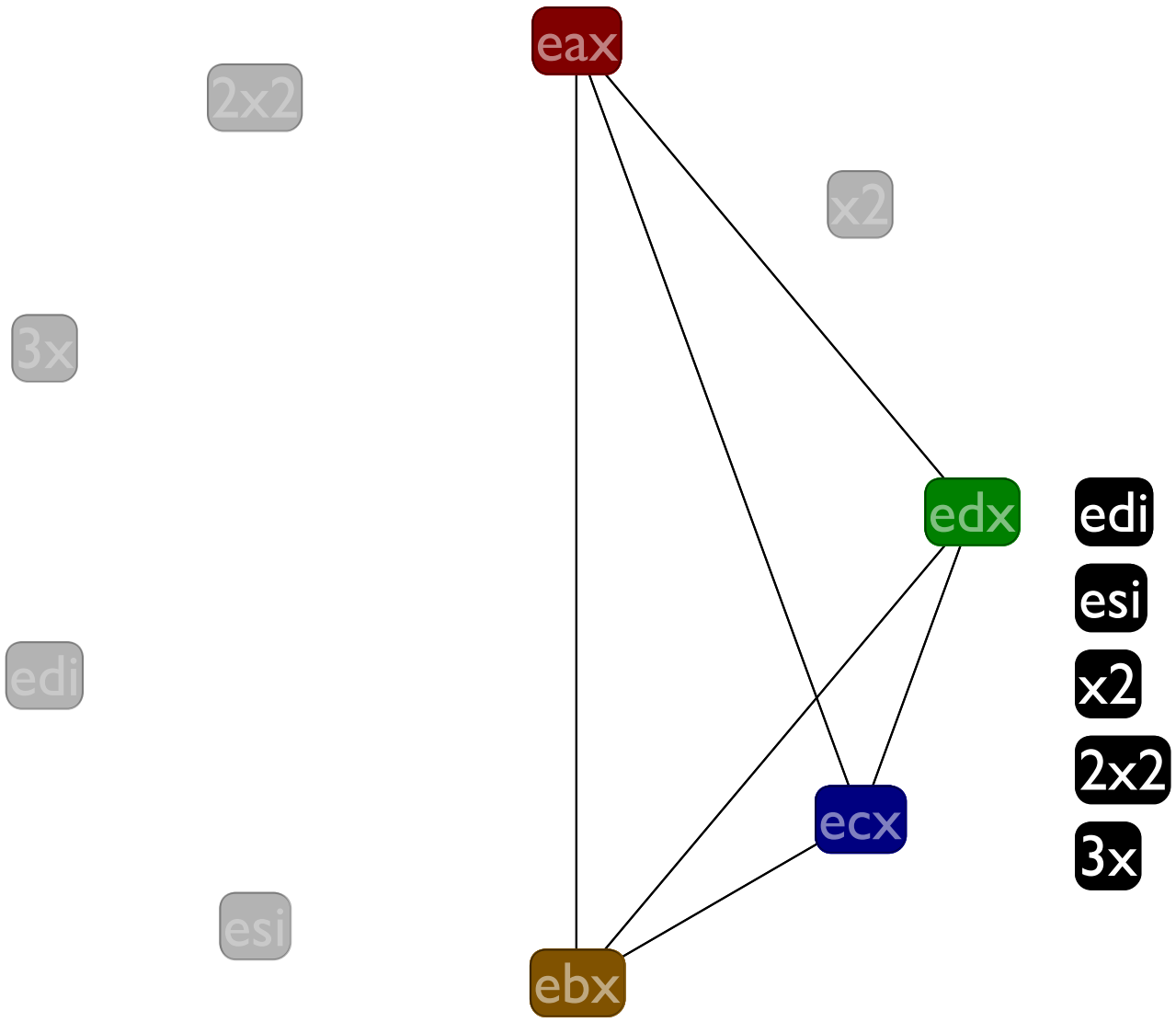


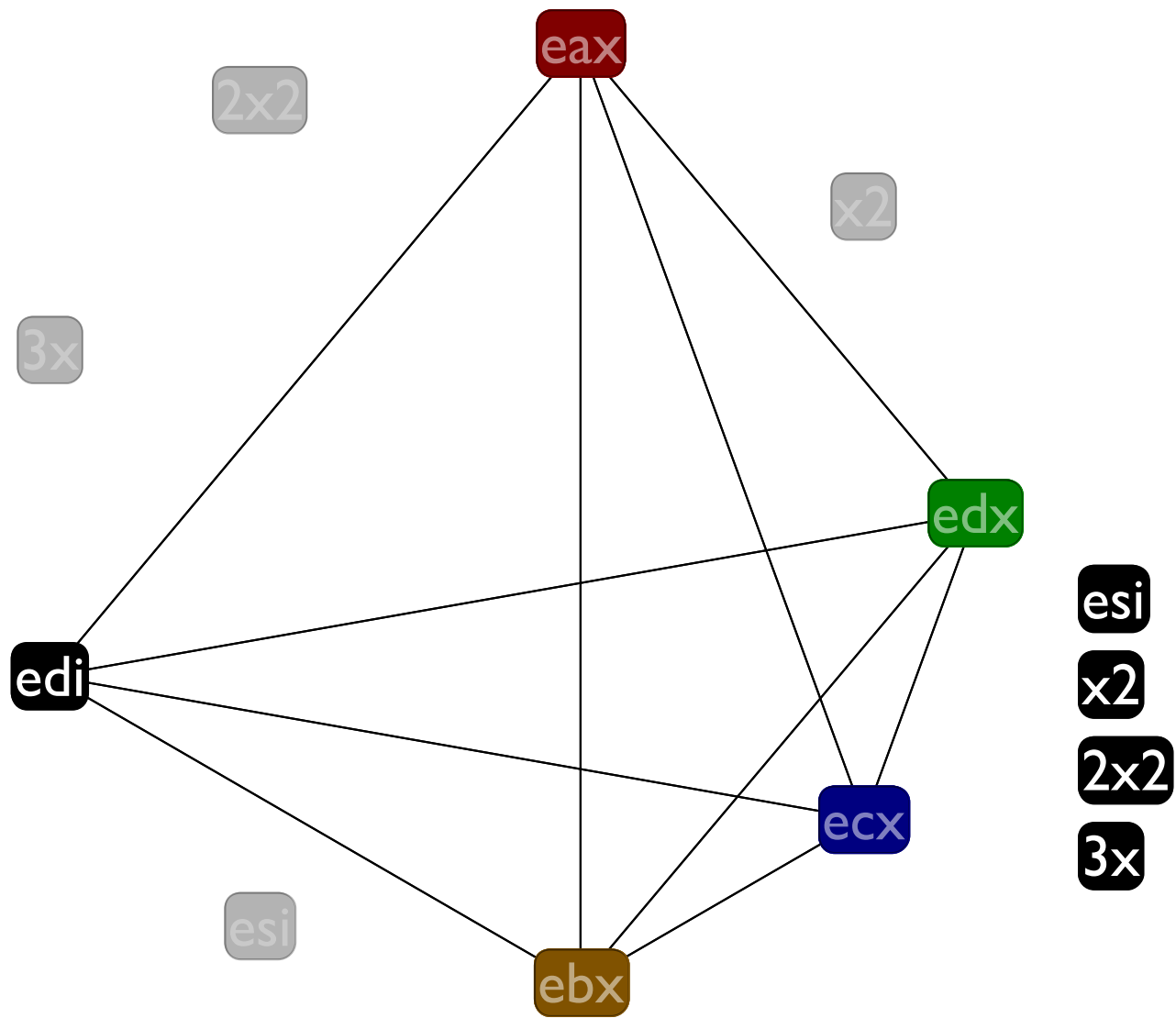


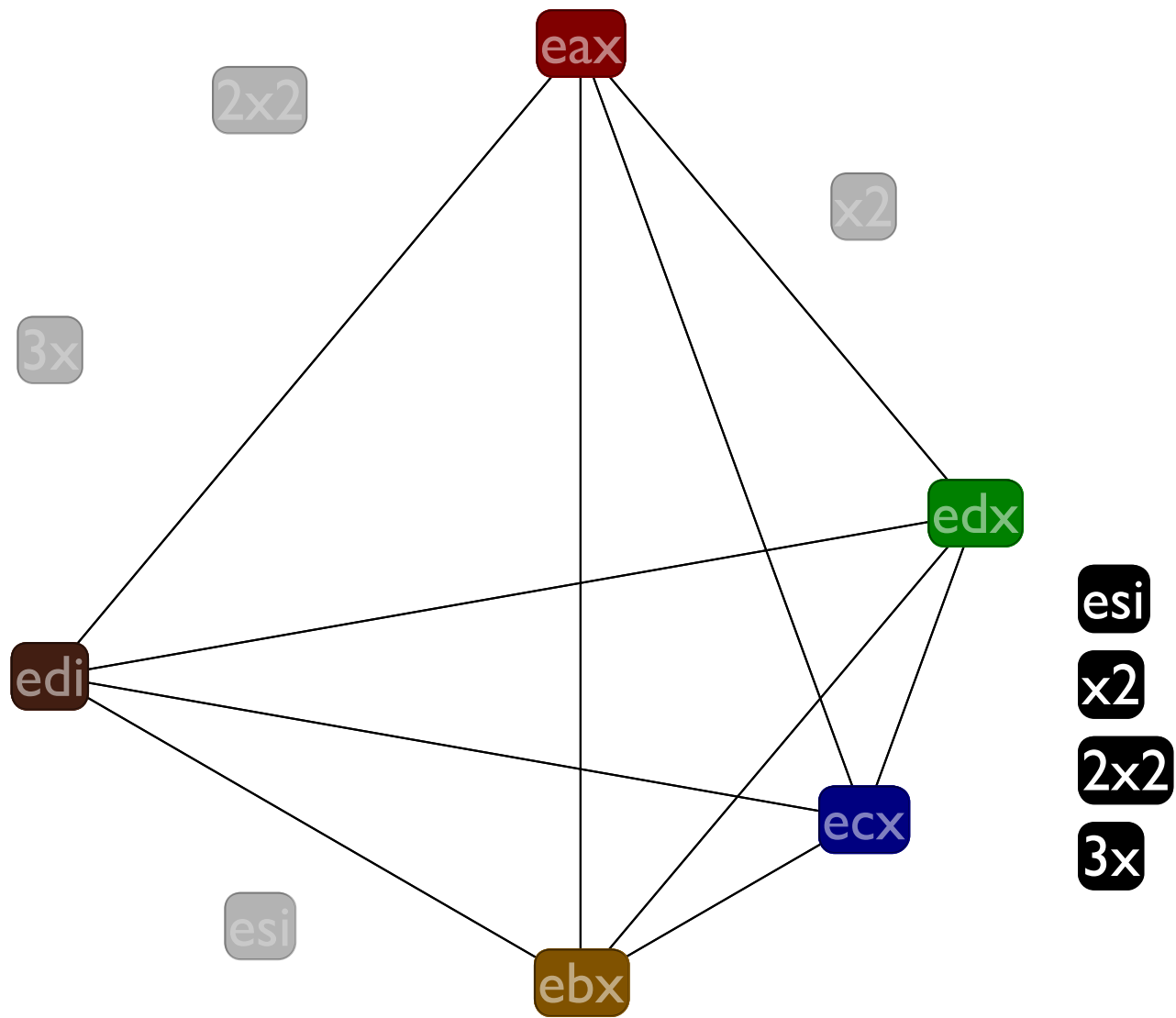




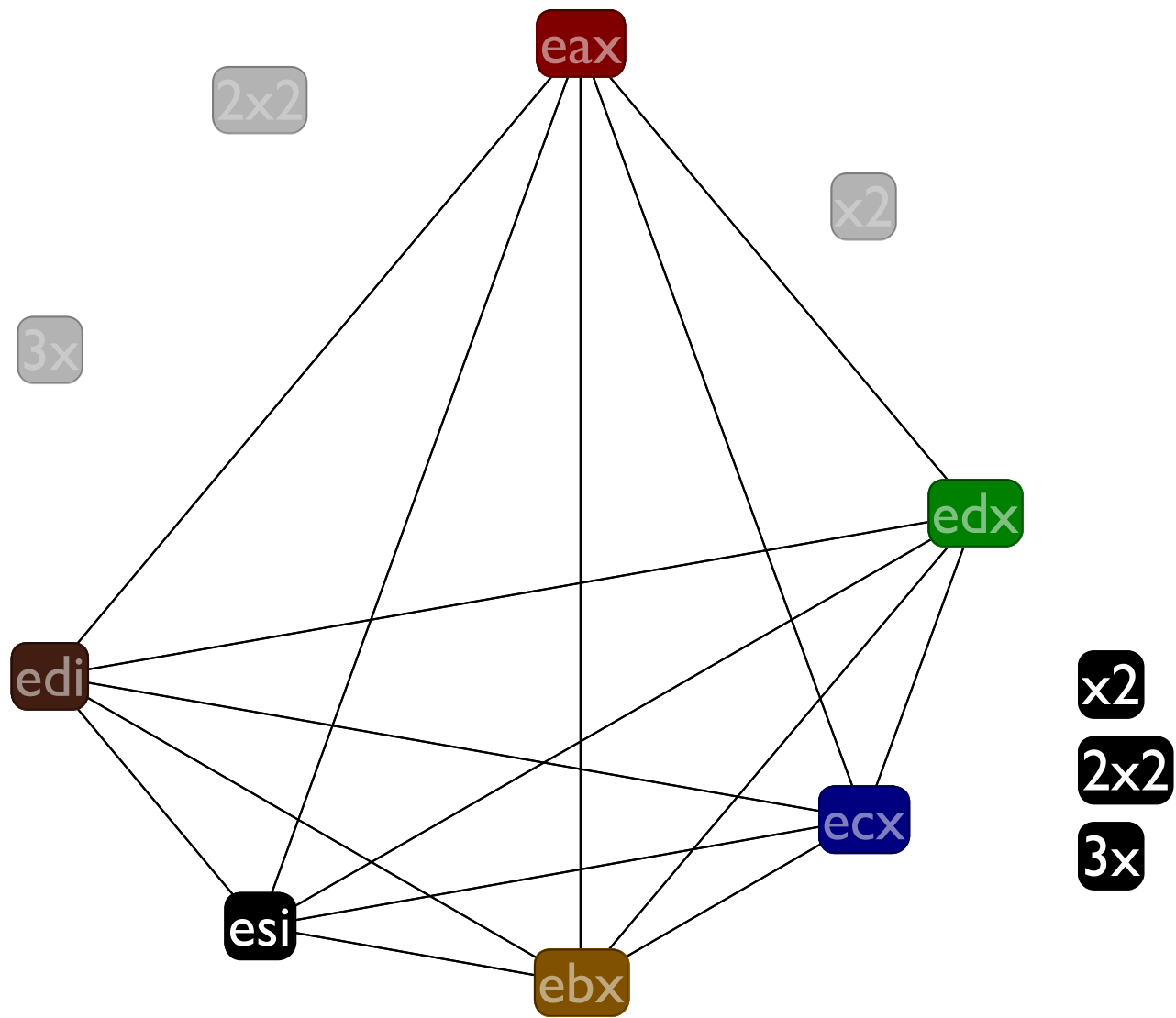


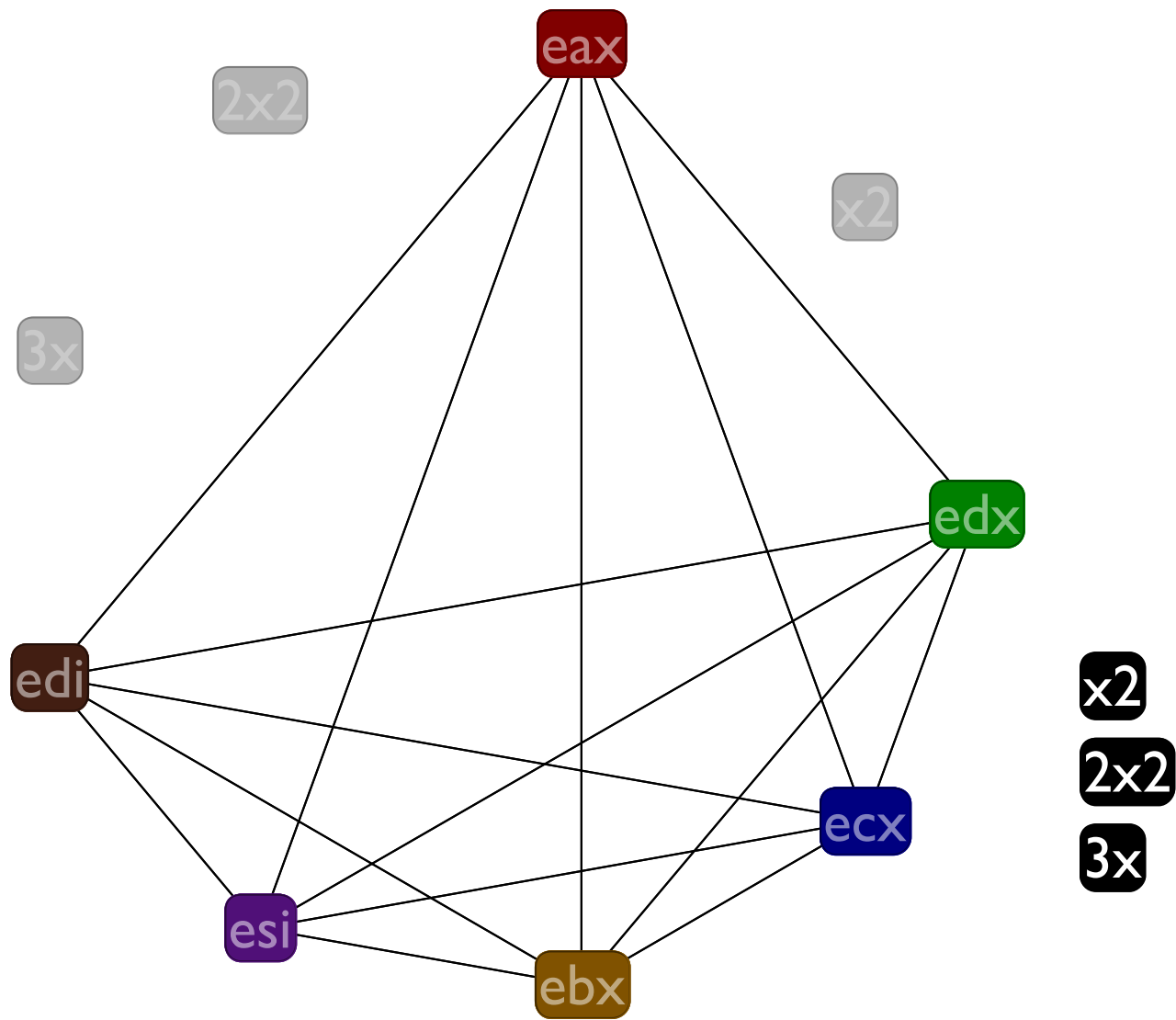


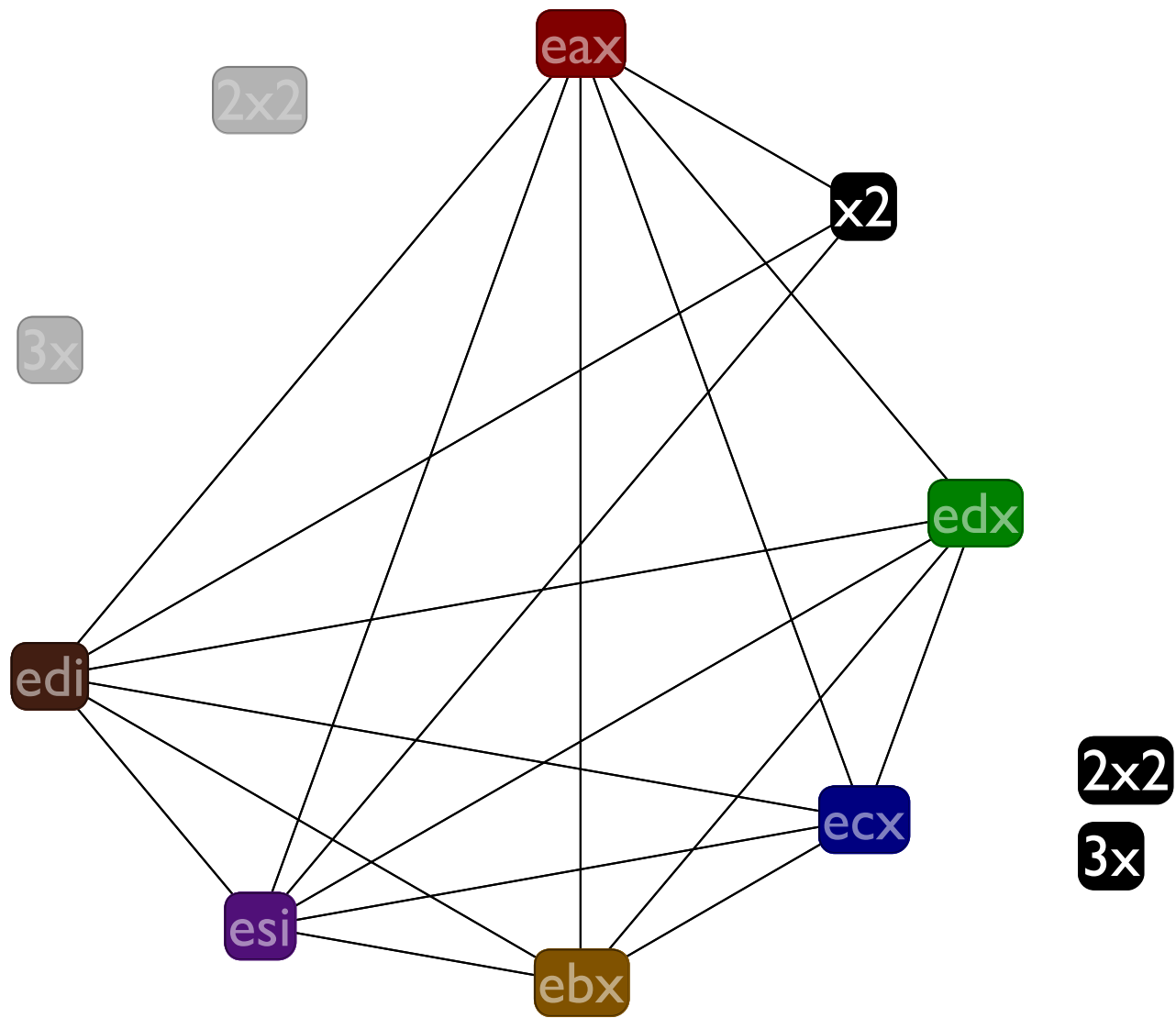


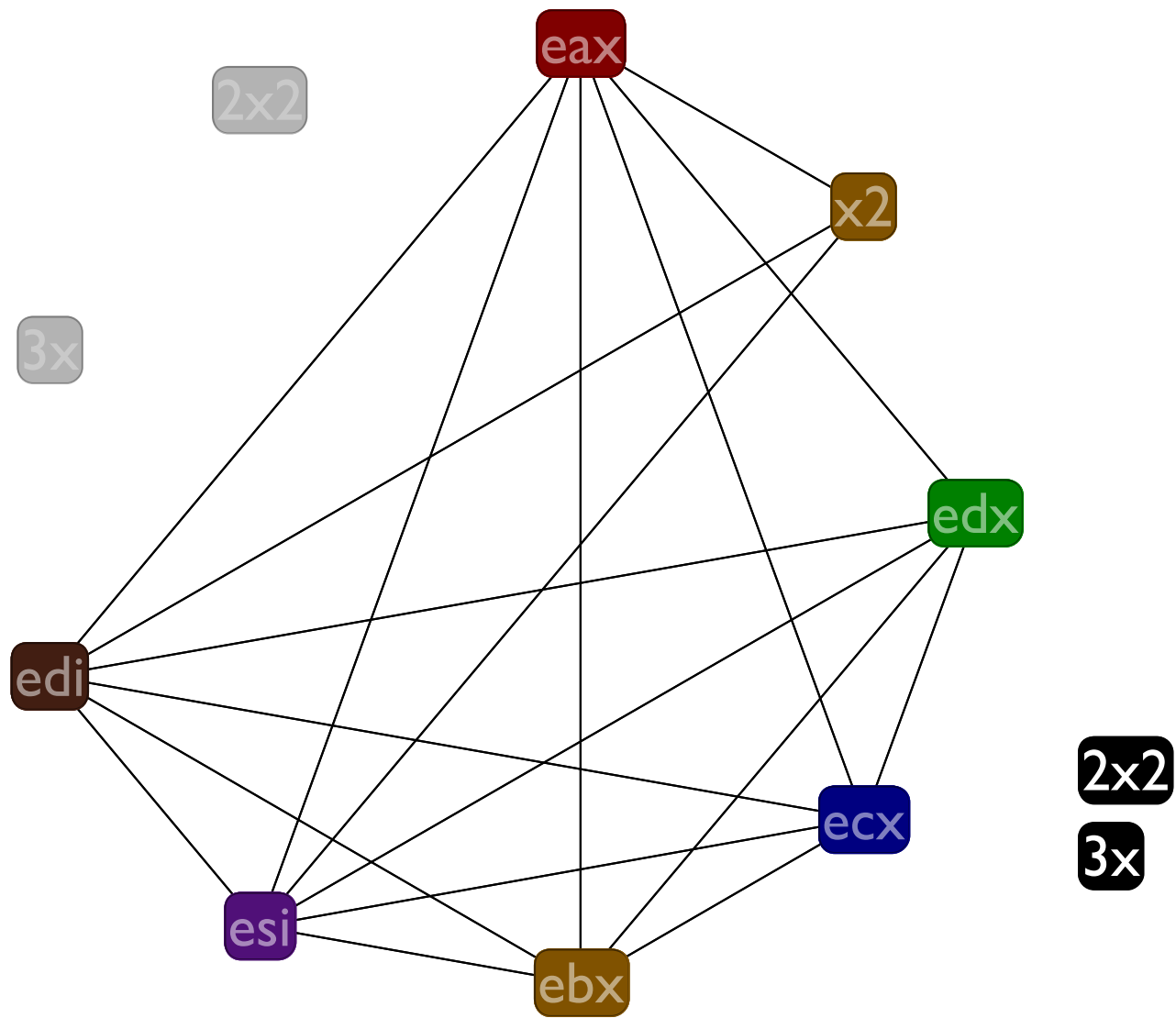


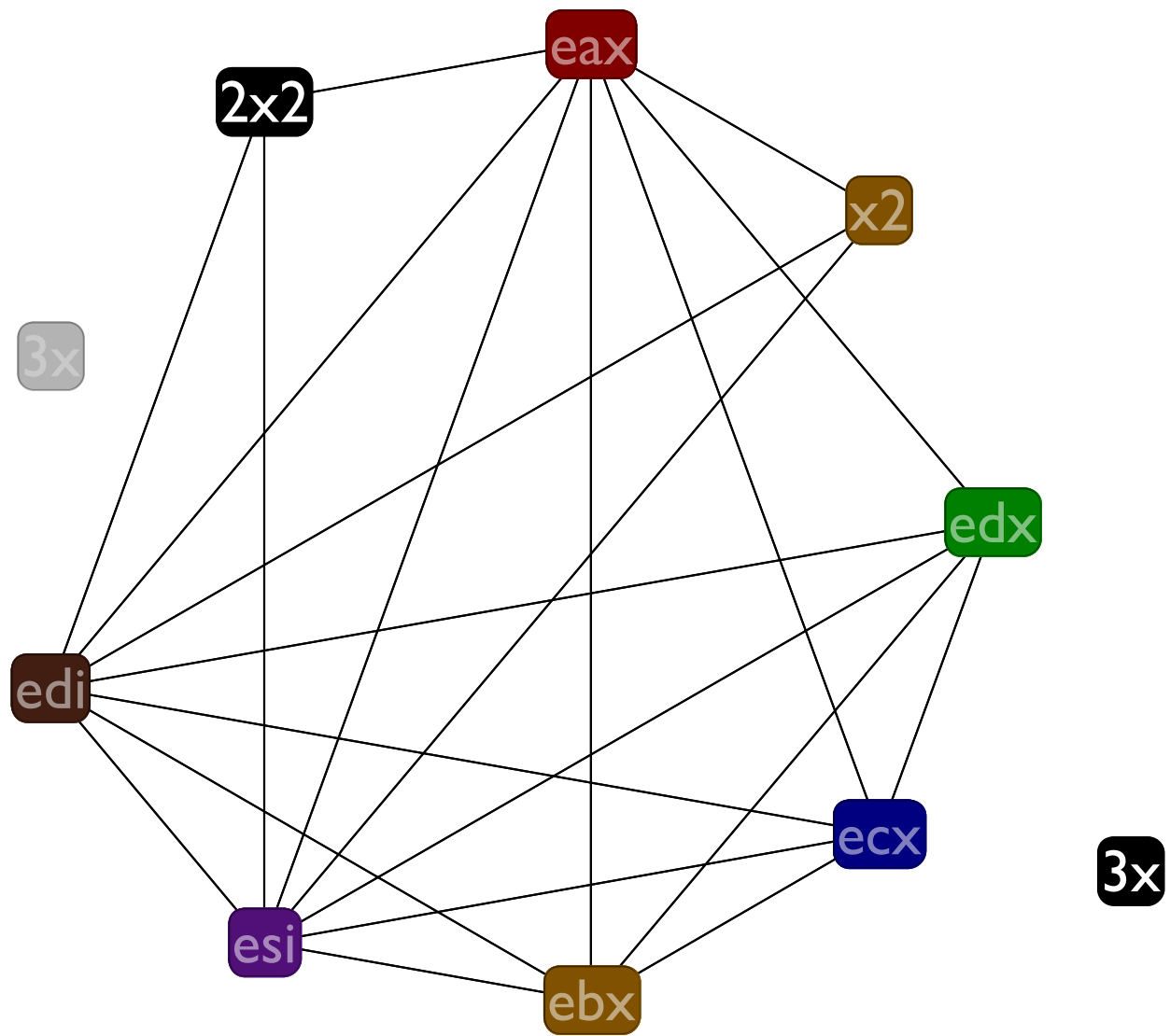


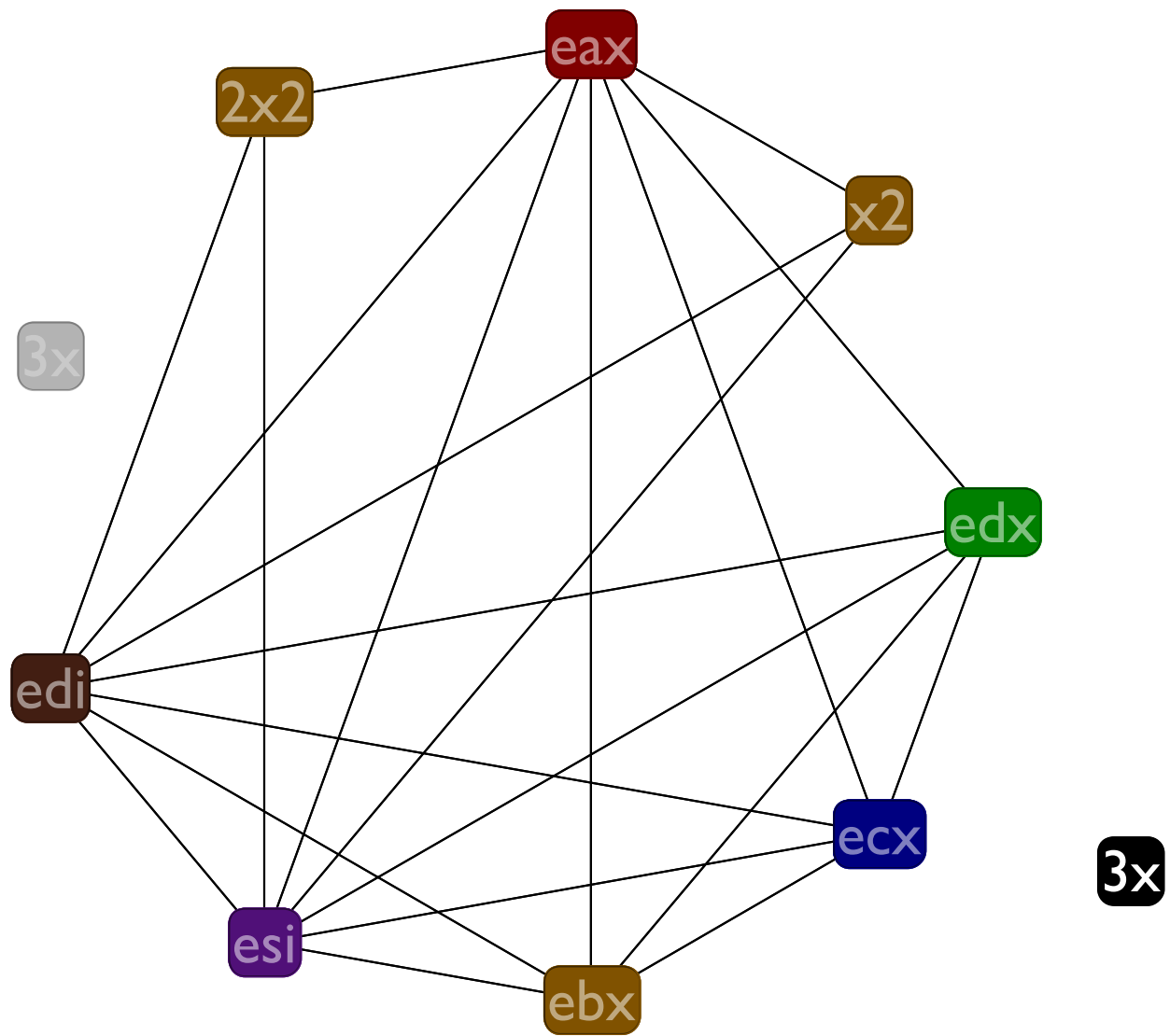


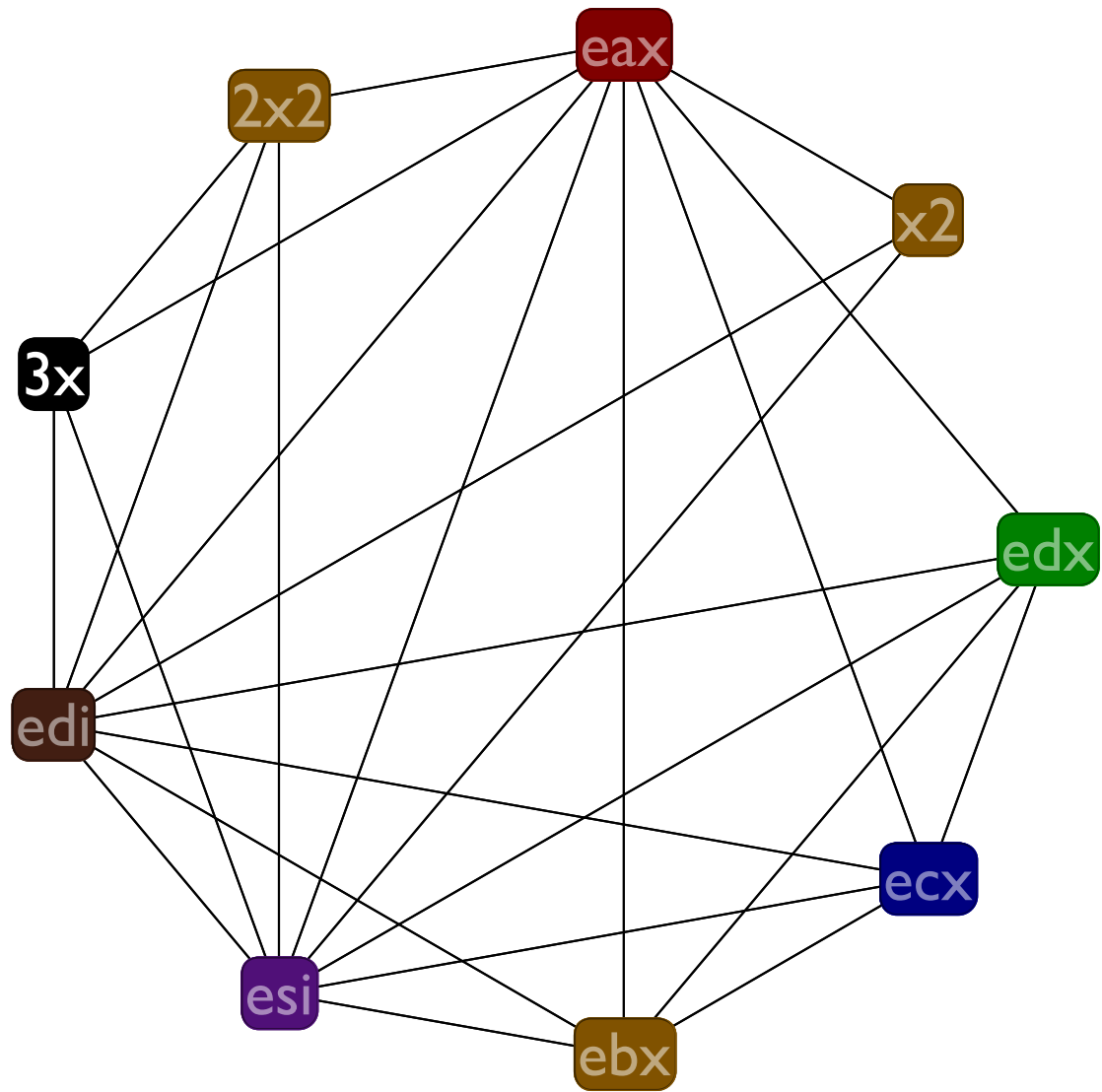


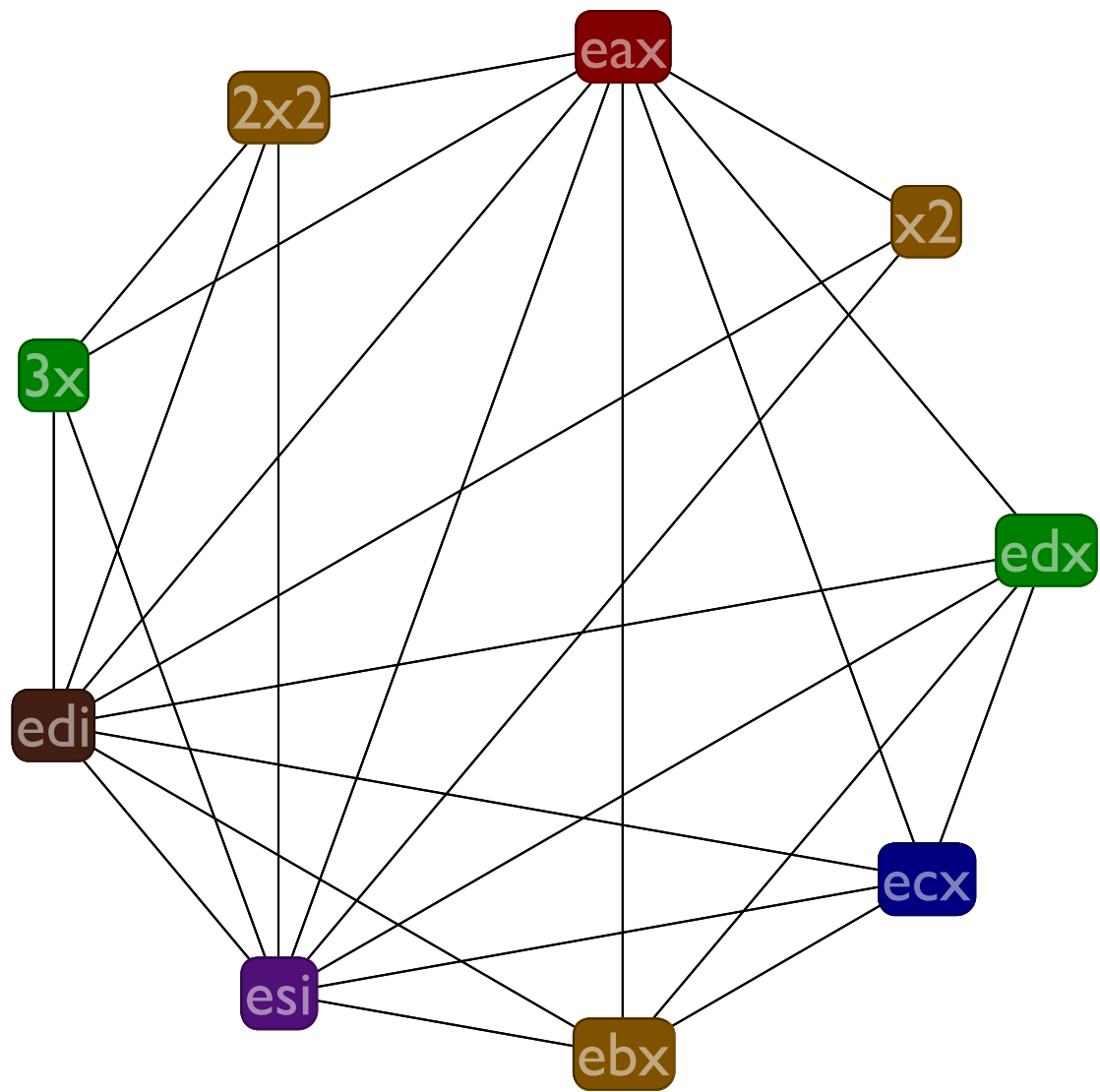














# An impossible to register-allocate program

```
:imp  
(eax <- 7)  
(ebx <- 7)  
(ecx <- 7)  
(edx <- 7)  
(edi <- 7)  
(esi <- 7)  
(x <- eax)  
(x += ebx)  
(x += ebx)  
(x += ecx)  
(x += edx)  
(x += edi)  
(x += esi)  
(x += eax)
```

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	()	()
6: <code>(edi &lt;- 7)</code>	()	()
7: <code>(esi &lt;- 7)</code>	()	()
8: <code>(x &lt;- eax)</code>	()	()
9: <code>(x += ebx)</code>	()	()
10: <code>(x += ebx)</code>	()	()
11: <code>(x += ecx)</code>	()	()
12: <code>(x += edx)</code>	()	()
13: <code>(x += edi)</code>	()	()
14: <code>(x += esi)</code>	()	()
15: <code>(x += eax)</code>	()	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	()	()
6: <code>(edi &lt;- 7)</code>	()	()
7: <code>(esi &lt;- 7)</code>	()	()
8: <code>(x &lt;- eax)</code>	(eax)	()
9: <code>(x += ebx)</code>	(ebx x)	()
10: <code>(x += ebx)</code>	(ebx x)	()
11: <code>(x += ecx)</code>	(ecx x)	()
12: <code>(x += edx)</code>	(edx x)	()
13: <code>(x += edi)</code>	(edi x)	()
14: <code>(x += esi)</code>	(esi x)	()
15: <code>(x += eax)</code>	(eax x)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	()	()
6: <code>(edi &lt;- 7)</code>	()	()
7: <code>(esi &lt;- 7)</code>	()	(eax)
8: <code>(x &lt;- eax)</code>	(eax)	(ebx x)
9: <code>(x += ebx)</code>	(ebx x)	(ebx x)
10: <code>(x += ebx)</code>	(ebx x)	(ecx x)
11: <code>(x += ecx)</code>	(ecx x)	(edx x)
12: <code>(x += edx)</code>	(edx x)	(edi x)
13: <code>(x += edi)</code>	(edi x)	(esi x)
14: <code>(x += esi)</code>	(esi x)	(eax x)
15: <code>(x += eax)</code>	(eax x)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	()	()
6: <code>(edi &lt;- 7)</code>	()	()
7: <code>(esi &lt;- 7)</code>	(eax)	(eax)
8: <code>(x &lt;- eax)</code>	(eax ebx)	(ebx x)
9: <code>(x += ebx)</code>	(ebx x)	(ebx x)
10: <code>(x += ebx)</code>	(ebx ecx x)	(ecx x)
11: <code>(x += ecx)</code>	(ecx edx x)	(edx x)
12: <code>(x += edx)</code>	(edi edx x)	(edi x)
13: <code>(x += edi)</code>	(edi esi x)	(esi x)
14: <code>(x += esi)</code>	(eax esi x)	(eax x)
15: <code>(x += eax)</code>	(eax x)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	()	()
6: <code>(edi &lt;- 7)</code>	()	(eax)
7: <code>(esi &lt;- 7)</code>	(eax)	(eax ebx)
8: <code>(x &lt;- eax)</code>	(eax ebx)	(ebx x)
9: <code>(x += ebx)</code>	(ebx x)	(ebx ecx x)
10: <code>(x += ebx)</code>	(ebx ecx x)	(ecx edx x)
11: <code>(x += ecx)</code>	(ecx edx x)	(edi edx x)
12: <code>(x += edx)</code>	(edi edx x)	(edi esi x)
13: <code>(x += edi)</code>	(edi esi x)	(eax esi x)
14: <code>(x += esi)</code>	(eax esi x)	(eax x)
15: <code>(x += eax)</code>	(eax x)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	()	()
6: <code>(edi &lt;- 7)</code>	(eax)	(eax)
7: <code>(esi &lt;- 7)</code>	(eax ebx)	(eax ebx)
8: <code>(x &lt;- eax)</code>	(eax ebx)	(ebx x)
9: <code>(x += ebx)</code>	(ebx ecx x)	(ebx ecx x)
10: <code>(x += ebx)</code>	(ebx ecx edx x)	(ecx edx x)
11: <code>(x += ecx)</code>	(ecx edi edx x)	(edi edx x)
12: <code>(x += edx)</code>	(edi edx esi x)	(edi esi x)
13: <code>(x += edi)</code>	(eax edi esi x)	(eax esi x)
14: <code>(x += esi)</code>	(eax esi x)	(eax x)
15: <code>(x += eax)</code>	(eax x)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	()	(eax)
6: <code>(edi &lt;- 7)</code>	(eax)	(eax ebx)
7: <code>(esi &lt;- 7)</code>	(eax ebx)	(eax ebx)
8: <code>(x &lt;- eax)</code>	(eax ebx)	(ebx ecx x)
9: <code>(x += ebx)</code>	(ebx ecx x)	(ebx ecx edx x)
10: <code>(x += ebx)</code>	(ebx ecx edx x)	(ecx edi edx x)
11: <code>(x += ecx)</code>	(ecx edi edx x)	(edi edx esi x)
12: <code>(x += edx)</code>	(edi edx esi x)	(eax edi esi x)
13: <code>(x += edi)</code>	(eax edi esi x)	(eax esi x)
14: <code>(x += esi)</code>	(eax esi x)	(eax x)
15: <code>(x += eax)</code>	(eax x)	()



# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	(eax)	(eax)
6: <code>(edi &lt;- 7)</code>	(eax ebx)	(eax ebx)
7: <code>(esi &lt;- 7)</code>	(eax ebx)	(eax ebx)
8: <code>(x &lt;- eax)</code>	(eax ebx ecx)	(ebx ecx x)
9: <code>(x += ebx)</code>	(ebx ecx edx x)	(ebx ecx edx x)
10: <code>(x += ebx)</code>	(ebx ecx edi edx x)	(ecx edi edx x)
11: <code>(x += ecx)</code>	(ecx edi edx esi x)	(edi edx esi x)
12: <code>(x += edx)</code>	(eax edi edx esi x)	(eax edi esi x)
13: <code>(x += edi)</code>	(eax edi esi x)	(eax esi x)
14: <code>(x += esi)</code>	(eax esi x)	(eax x)
15: <code>(x += eax)</code>	(eax x)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	(eax)
5: <code>(edx &lt;- 7)</code>	(eax)	(eax ebx)
6: <code>(edi &lt;- 7)</code>	(eax ebx)	(eax ebx)
7: <code>(esi &lt;- 7)</code>	(eax ebx)	(eax ebx ecx)
8: <code>(x &lt;- eax)</code>	(eax ebx ecx)	(ebx ecx edx x)
9: <code>(x += ebx)</code>	(ebx ecx edx x)	(ebx ecx edi edx x)
10: <code>(x += ebx)</code>	(ebx ecx edi edx x)	(ecx edi edx esi x)
11: <code>(x += ecx)</code>	(ecx edi edx esi x)	(eax edi edx esi x)
12: <code>(x += edx)</code>	(eax edi edx esi x)	(eax edi esi x)
13: <code>(x += edi)</code>	(eax edi esi x)	(eax esi x)
14: <code>(x += esi)</code>	(eax esi x)	(eax x)
15: <code>(x += eax)</code>	(eax x)	()

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>()</code>
2: <code>(eax &lt;- 7)</code>	<code>()</code>	<code>()</code>
3: <code>(ebx &lt;- 7)</code>	<code>()</code>	<code>()</code>
4: <code>(ecx &lt;- 7)</code>	<code>(eax)</code>	<code>(eax)</code>
5: <code>(edx &lt;- 7)</code>	<code>(eax ebx)</code>	<code>(eax ebx)</code>
6: <code>(edi &lt;- 7)</code>	<code>(eax ebx)</code>	<code>(eax ebx)</code>
7: <code>(esi &lt;- 7)</code>	<code>(eax ebx ecx)</code>	<code>(eax ebx ecx)</code>
8: <code>(x &lt;- eax)</code>	<code>(eax ebx ecx edx)</code>	<code>(ebx ecx edx x)</code>
9: <code>(x += ebx)</code>	<code>(ebx ecx edi edx x)</code>	<code>(ebx ecx edi edx x)</code>
10: <code>(x += ebx)</code>	<code>(ebx ecx edi edx esi x)</code>	<code>(ecx edi edx esi x)</code>
11: <code>(x += ecx)</code>	<code>(eax ecx edi edx esi x)</code>	<code>(eax edi edx esi x)</code>
12: <code>(x += edx)</code>	<code>(eax edi edx esi x)</code>	<code>(eax edi esi x)</code>
13: <code>(x += edi)</code>	<code>(eax edi esi x)</code>	<code>(eax esi x)</code>
14: <code>(x += esi)</code>	<code>(eax esi x)</code>	<code>(eax x)</code>
15: <code>(x += eax)</code>	<code>(eax x)</code>	<code>()</code>

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>()</code>
2: <code>(eax &lt;- 7)</code>	<code>()</code>	<code>()</code>
3: <code>(ebx &lt;- 7)</code>	<code>()</code>	<code>(eax)</code>
4: <code>(ecx &lt;- 7)</code>	<code>(eax)</code>	<code>(eax ebx)</code>
5: <code>(edx &lt;- 7)</code>	<code>(eax ebx)</code>	<code>(eax ebx)</code>
6: <code>(edi &lt;- 7)</code>	<code>(eax ebx)</code>	<code>(eax ebx ecx)</code>
7: <code>(esi &lt;- 7)</code>	<code>(eax ebx ecx)</code>	<code>(eax ebx ecx edx)</code>
8: <code>(x &lt;- eax)</code>	<code>(eax ebx ecx edx)</code>	<code>(ebx ecx edi edx x)</code>
9: <code>(x += ebx)</code>	<code>(ebx ecx edi edx x)</code>	<code>(ebx ecx edi edx esi x)</code>
10: <code>(x += ebx)</code>	<code>(ebx ecx edi edx esi x)</code>	<code>(eax ecx edi edx esi x)</code>
11: <code>(x += ecx)</code>	<code>(eax ecx edi edx esi x)</code>	<code>(eax edi edx esi x)</code>
12: <code>(x += edx)</code>	<code>(eax edi edx esi x)</code>	<code>(eax edi esi x)</code>
13: <code>(x += edi)</code>	<code>(eax edi esi x)</code>	<code>(eax esi x)</code>
14: <code>(x += esi)</code>	<code>(eax esi x)</code>	<code>(eax x)</code>
15: <code>(x += eax)</code>	<code>(eax x)</code>	<code>()</code>

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>()</code>
2: <code>(eax &lt;- 7)</code>	<code>()</code>	<code>()</code>
3: <code>(ebx &lt;- 7)</code>	<code>(eax)</code>	<code>(eax)</code>
4: <code>(ecx &lt;- 7)</code>	<code>(eax ebx)</code>	<code>(eax ebx)</code>
5: <code>(edx &lt;- 7)</code>	<code>(eax ebx)</code>	<code>(eax ebx)</code>
6: <code>(edi &lt;- 7)</code>	<code>(eax ebx ecx)</code>	<code>(eax ebx ecx)</code>
7: <code>(esi &lt;- 7)</code>	<code>(eax ebx ecx edx)</code>	<code>(eax ebx ecx edx)</code>
8: <code>(x &lt;- eax)</code>	<code>(eax ebx ecx edi edx)</code>	<code>(ebx ecx edi edx x)</code>
9: <code>(x += ebx)</code>	<code>(ebx ecx edi edx esi x)</code>	<code>(ebx ecx edi edx esi x)</code>
10: <code>(x += ebx)</code>	<code>(eax ebx ecx edi edx esi x)</code>	<code>(eax ecx edi edx esi x)</code>
11: <code>(x += ecx)</code>	<code>(eax ecx edi edx esi x)</code>	<code>(eax edi edx esi x)</code>
12: <code>(x += edx)</code>	<code>(eax edi edx esi x)</code>	<code>(eax edi esi x)</code>
13: <code>(x += edi)</code>	<code>(eax edi esi x)</code>	<code>(eax esi x)</code>
14: <code>(x += esi)</code>	<code>(eax esi x)</code>	<code>(eax x)</code>
15: <code>(x += eax)</code>	<code>(eax x)</code>	<code>()</code>

# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	(eax)
3: <code>(ebx &lt;- 7)</code>	(eax)	(eax ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebx)	(eax ebx)
5: <code>(edx &lt;- 7)</code>	(eax ebx)	(eax ebx ecx)
6: <code>(edi &lt;- 7)</code>	(eax ebx ecx)	(eax ebx ecx edx)
7: <code>(esi &lt;- 7)</code>	(eax ebx ecx edx)	(eax ebx ecx edi edx)
8: <code>(x &lt;- eax)</code>	(eax ebx ecx edi edx)	(ebx ecx edi edx esi x)
9: <code>(x += ebx)</code>	(ebx ecx edi edx esi x)	(eax ebx ecx edi edx esi x)
10: <code>(x += ebx)</code>	(eax ebx ecx edi edx esi x)	(eax ecx edi edx esi x)
11: <code>(x += ecx)</code>	(eax ecx edi edx esi x)	(eax edi edx esi x)
12: <code>(x += edx)</code>	(eax edi edx esi x)	(eax edi esi x)
13: <code>(x += edi)</code>	(eax edi esi x)	(eax esi x)
14: <code>(x += esi)</code>	(eax esi x)	(eax x)
15: <code>(x += eax)</code>	(eax x)	()

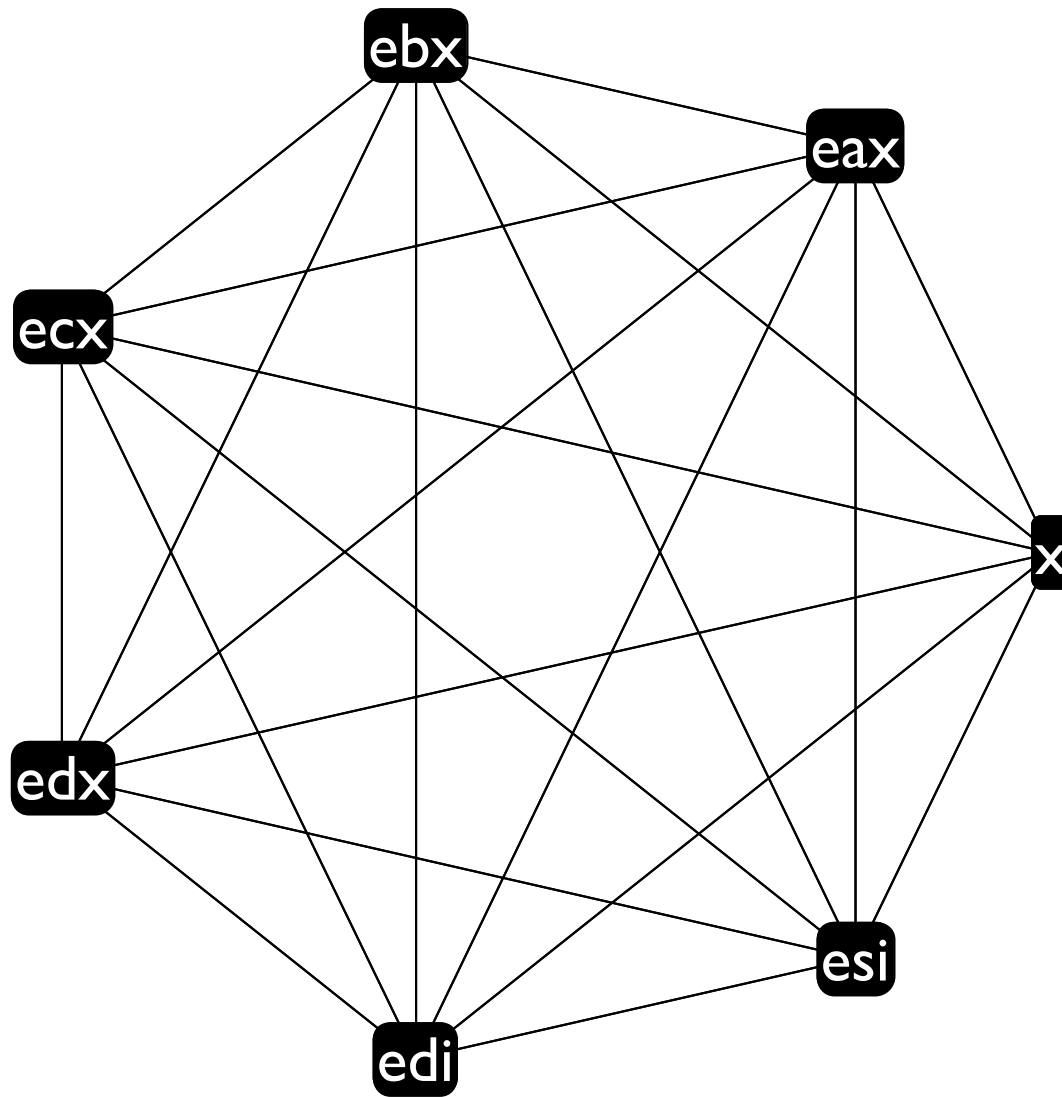
# Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>()</code>
2: <code>(eax &lt;- 7)</code>	<code>()</code>	<code>(eax)</code>
3: <code>(ebx &lt;- 7)</code>	<code>(eax)</code>	<code>(eax ebx)</code>
4: <code>(ecx &lt;- 7)</code>	<code>(eax ebx)</code>	<code>(eax ebx)</code>
5: <code>(edx &lt;- 7)</code>	<code>(eax ebx ecx)</code>	<code>(eax ebx ecx)</code>
6: <code>(edi &lt;- 7)</code>	<code>(eax ebx ecx edx)</code>	<code>(eax ebx ecx edx)</code>
7: <code>(esi &lt;- 7)</code>	<code>(eax ebx ecx edi edx)</code>	<code>(eax ebx ecx edi edx)</code>
8: <code>(x &lt;- eax)</code>	<code>(eax ebx ecx edi edx esi)</code>	<code>(ebx ecx edi edx esi x)</code>
9: <code>(x += ebx)</code>	<code>(eax ebx ecx edi edx esi x)</code>	<code>(eax ebx ecx edi edx esi x)</code>
10: <code>(x += ebx)</code>	<code>(eax ebx ecx edi edx esi x)</code>	<code>(eax ecx edi edx esi x)</code>
11: <code>(x += ecx)</code>	<code>(eax ecx edi edx esi x)</code>	<code>(eax edi edx esi x)</code>
12: <code>(x += edx)</code>	<code>(eax edi edx esi x)</code>	<code>(eax edi esi x)</code>
13: <code>(x += edi)</code>	<code>(eax edi esi x)</code>	<code>(eax esi x)</code>
14: <code>(x += esi)</code>	<code>(eax esi x)</code>	<code>(eax x)</code>
15: <code>(x += eax)</code>	<code>(eax x)</code>	<code>()</code>

# Liveness

	<b>in</b>	<b>out</b>
1: <b>:imp</b>	()	()
2: <b>(eax &lt;- 7)</b>	()	(eax)
3: <b>(ebx &lt;- 7)</b>	(eax)	(eax ebx)
4: <b>(ecx &lt;- 7)</b>	(eax ebx)	(eax ebx ecx)
5: <b>(edx &lt;- 7)</b>	(eax ebx ecx)	(eax ebx ecx edx)
6: <b>(edi &lt;- 7)</b>	(eax ebx ecx edx)	(eax ebx ecx edi edx)
7: <b>(esi &lt;- 7)</b>	(eax ebx ecx edi edx)	(eax ebx ecx edi edx esi)
8: <b>(x &lt;- eax)</b>	(eax ebx ecx edi edx esi)	(eax ebx ecx edi edx esi x)
9: <b>(x += ebx)</b>	(eax ebx ecx edi edx esi x)	(eax ebx ecx edi edx esi x)
10: <b>(x += ebx)</b>	(eax ebx ecx edi edx esi x)	(eax ecx edi edx esi x)
11: <b>(x += ecx)</b>	(eax ecx edi edx esi x)	(eax edi edx esi x)
12: <b>(x += edx)</b>	(eax edi edx esi x)	(eax edi esi x)
13: <b>(x += edi)</b>	(eax edi esi x)	(eax esi x)
14: <b>(x += esi)</b>	(eax esi x)	(eax x)
15: <b>(x += eax)</b>	(eax x)	()





# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	()	()
6: <code>(edi &lt;- 7)</code>	()	()
7: <code>(esi &lt;- 7)</code>	()	()
8: <code>((mem ebp -4) &lt;- eax)</code>	()	()
9: <code>(s0 &lt;- (mem ebp -4))</code>	()	()
10: <code>(s0 += ebx)</code>	()	()
11: <code>((mem ebp -4) &lt;- s0)</code>	()	()
12: <code>(s1 &lt;- (mem ebp -4))</code>	()	()
13: <code>(s1 += ebx)</code>	()	()
14: <code>((mem ebp -4) &lt;- s1)</code>	()	()
15: <code>(s2 &lt;- (mem ebp -4))</code>	()	()
16: <code>(s2 += ecx)</code>	()	()
17: <code>((mem ebp -4) &lt;- s2)</code>	()	()
18: <code>(s3 &lt;- (mem ebp -4))</code>	()	()
19: <code>(s3 += edx)</code>	()	()
20: <code>((mem ebp -4) &lt;- s3)</code>	()	()
21: <code>(s4 &lt;- (mem ebp -4))</code>	()	()
22: <code>(s4 += edi)</code>	()	()
23: <code>((mem ebp -4) &lt;- s4)</code>	()	()
24: <code>(s5 &lt;- (mem ebp -4))</code>	()	()
25: <code>(s5 += esi)</code>	()	()
26: <code>((mem ebp -4) &lt;- s5)</code>	()	()
27: <code>(s6 &lt;- (mem ebp -4))</code>	()	()
28: <code>(s6 += eax)</code>	()	()
29: <code>((mem ebp -4) &lt;- s6)</code>	()	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	()	()
6: <code>(edi &lt;- 7)</code>	()	()
7: <code>(esi &lt;- 7)</code>	()	()
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp)	()
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp)	()
10: <code>(s0 += ebx)</code>	(ebx s0)	()
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp s0)	()
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp)	()
13: <code>(s1 += ebx)</code>	(ebx s1)	()
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp s1)	()
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp)	()
16: <code>(s2 += ecx)</code>	(ecx s2)	()
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp s2)	()
18: <code>(s3 &lt;- (mem ebp -4))</code>	(ebp)	()
19: <code>(s3 += edx)</code>	(edx s3)	()
20: <code>((mem ebp -4) &lt;- s3)</code>	(ebp s3)	()
21: <code>(s4 &lt;- (mem ebp -4))</code>	(ebp)	()
22: <code>(s4 += edi)</code>	(edi s4)	()
23: <code>((mem ebp -4) &lt;- s4)</code>	(ebp s4)	()
24: <code>(s5 &lt;- (mem ebp -4))</code>	(ebp)	()
25: <code>(s5 += esi)</code>	(esi s5)	()
26: <code>((mem ebp -4) &lt;- s5)</code>	(ebp s5)	()
27: <code>(s6 &lt;- (mem ebp -4))</code>	(ebp)	()
28: <code>(s6 += eax)</code>	(eax s6)	()
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>()</code>
2: <code>(eax &lt;- 7)</code>	<code>()</code>	<code>()</code>
3: <code>(ebx &lt;- 7)</code>	<code>()</code>	<code>()</code>
4: <code>(ecx &lt;- 7)</code>	<code>()</code>	<code>()</code>
5: <code>(edx &lt;- 7)</code>	<code>()</code>	<code>()</code>
6: <code>(edi &lt;- 7)</code>	<code>()</code>	<code>()</code>
7: <code>(esi &lt;- 7)</code>	<code>()</code>	<code>(eax ebp)</code>
8: <code>((mem ebp -4) &lt;- eax)</code>	<code>(eax ebp)</code>	<code>(ebp)</code>
9: <code>(s0 &lt;- (mem ebp -4))</code>	<code>(ebp)</code>	<code>(ebx s0)</code>
10: <code>(s0 += ebx)</code>	<code>(ebx s0)</code>	<code>(ebp s0)</code>
11: <code>((mem ebp -4) &lt;- s0)</code>	<code>(ebp s0)</code>	<code>(ebp)</code>
12: <code>(s1 &lt;- (mem ebp -4))</code>	<code>(ebp)</code>	<code>(ebx s1)</code>
13: <code>(s1 += ebx)</code>	<code>(ebx s1)</code>	<code>(ebp s1)</code>
14: <code>((mem ebp -4) &lt;- s1)</code>	<code>(ebp s1)</code>	<code>(ebp)</code>
15: <code>(s2 &lt;- (mem ebp -4))</code>	<code>(ebp)</code>	<code>(ecx s2)</code>
16: <code>(s2 += ecx)</code>	<code>(ecx s2)</code>	<code>(ebp s2)</code>
17: <code>((mem ebp -4) &lt;- s2)</code>	<code>(ebp s2)</code>	<code>(ebp)</code>
18: <code>(s3 &lt;- (mem ebp -4))</code>	<code>(ebp)</code>	<code>(edx s3)</code>
19: <code>(s3 += edx)</code>	<code>(edx s3)</code>	<code>(ebp s3)</code>
20: <code>((mem ebp -4) &lt;- s3)</code>	<code>(ebp s3)</code>	<code>(ebp)</code>
21: <code>(s4 &lt;- (mem ebp -4))</code>	<code>(ebp)</code>	<code>(edi s4)</code>
22: <code>(s4 += edi)</code>	<code>(edi s4)</code>	<code>(ebp s4)</code>
23: <code>((mem ebp -4) &lt;- s4)</code>	<code>(ebp s4)</code>	<code>(ebp)</code>
24: <code>(s5 &lt;- (mem ebp -4))</code>	<code>(ebp)</code>	<code>(esi s5)</code>
25: <code>(s5 += esi)</code>	<code>(esi s5)</code>	<code>(ebp s5)</code>
26: <code>((mem ebp -4) &lt;- s5)</code>	<code>(ebp s5)</code>	<code>(ebp)</code>
27: <code>(s6 &lt;- (mem ebp -4))</code>	<code>(ebp)</code>	<code>(eax s6)</code>
28: <code>(s6 += eax)</code>	<code>(eax s6)</code>	<code>(ebp s6)</code>
29: <code>((mem ebp -4) &lt;- s6)</code>	<code>(ebp s6)</code>	<code>()</code>

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	()	()
6: <code>(edi &lt;- 7)</code>	()	()
7: <code>(esi &lt;- 7)</code>	(eax ebp)	(eax ebp)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp)	(ebp)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx)	(ebx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx s0)	(ebp s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp s0)	(ebp)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx)	(ebx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx s1)	(ebp s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp s1)	(ebp)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx)	(ecx s2)
16: <code>(s2 += ecx)</code>	(ebp ecx s2)	(ebp s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp s2)	(ebp)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(ebp edx)	(edx s3)
19: <code>(s3 += edx)</code>	(ebp edx s3)	(ebp s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(ebp s3)	(ebp)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(ebp edi)	(edi s4)
22: <code>(s4 += edi)</code>	(ebp edi s4)	(ebp s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(ebp s4)	(ebp)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(ebp esi)	(esi s5)
25: <code>(s5 += esi)</code>	(ebp esi s5)	(ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(ebp s5)	(ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	()	()
6: <code>(edi &lt;- 7)</code>	()	(eax ebp)
7: <code>(esi &lt;- 7)</code>	(eax ebp)	(eax ebp)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp)	(ebp ebx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx)	(ebp ebx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx s0)	(ebp s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp s0)	(ebp ebx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx)	(ebp ebx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx s1)	(ebp s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp s1)	(ebp ecx)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx)	(ebp ecx s2)
16: <code>(s2 += ecx)</code>	(ebp ecx s2)	(ebp s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp s2)	(ebp edx)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(ebp edx)	(ebp edx s3)
19: <code>(s3 += edx)</code>	(ebp edx s3)	(ebp s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(ebp s3)	(ebp edi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(ebp edi)	(ebp edi s4)
22: <code>(s4 += edi)</code>	(ebp edi s4)	(ebp s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(ebp s4)	(ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(ebp esi)	(ebp esi s5)
25: <code>(s5 += esi)</code>	(ebp esi s5)	(ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	()	()
5: <code>(edx &lt;- 7)</code>	()	()
6: <code>(edi &lt;- 7)</code>	(eax ebp)	(eax ebp)
7: <code>(esi &lt;- 7)</code>	(eax ebp)	(eax ebp)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx)	(ebp ebx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx)	(ebp ebx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx s0)	(ebp s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx s0)	(ebp ebx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx)	(ebp ebx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx s1)	(ebp s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx s1)	(ebp ecx)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx)	(ebp ecx s2)
16: <code>(s2 += ecx)</code>	(ebp ecx s2)	(ebp s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp edx s2)	(ebp edx)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(ebp edx)	(ebp edx s3)
19: <code>(s3 += edx)</code>	(ebp edx s3)	(ebp s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(ebp edi s3)	(ebp edi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(ebp edi)	(ebp edi s4)
22: <code>(s4 += edi)</code>	(ebp edi s4)	(ebp s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(ebp esi s4)	(ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(ebp esi)	(ebp esi s5)
25: <code>(s5 += esi)</code>	(ebp esi s5)	(ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>()</code>
2: <code>(eax &lt;- 7)</code>	<code>()</code>	<code>()</code>
3: <code>(ebx &lt;- 7)</code>	<code>()</code>	<code>()</code>
4: <code>(ecx &lt;- 7)</code>	<code>()</code>	<code>()</code>
5: <code>(edx &lt;- 7)</code>	<code>()</code>	<code>(eax ebp)</code>
6: <code>(edi &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp)</code>
7: <code>(esi &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp ebx)</code>
8: <code>((mem ebp -4) &lt;- eax)</code>	<code>(eax ebp ebx)</code>	<code>(ebp ebx)</code>
9: <code>(s0 &lt;- (mem ebp -4))</code>	<code>(ebp ebx)</code>	<code>(ebp ebx s0)</code>
10: <code>(s0 += ebx)</code>	<code>(ebp ebx s0)</code>	<code>(ebp ebx s0)</code>
11: <code>((mem ebp -4) &lt;- s0)</code>	<code>(ebp ebx s0)</code>	<code>(ebp ebx)</code>
12: <code>(s1 &lt;- (mem ebp -4))</code>	<code>(ebp ebx)</code>	<code>(ebp ebx s1)</code>
13: <code>(s1 += ebx)</code>	<code>(ebp ebx s1)</code>	<code>(ebp ecx s1)</code>
14: <code>((mem ebp -4) &lt;- s1)</code>	<code>(ebp ecx s1)</code>	<code>(ebp ecx)</code>
15: <code>(s2 &lt;- (mem ebp -4))</code>	<code>(ebp ecx)</code>	<code>(ebp ecx s2)</code>
16: <code>(s2 += ecx)</code>	<code>(ebp ecx s2)</code>	<code>(ebp edx s2)</code>
17: <code>((mem ebp -4) &lt;- s2)</code>	<code>(ebp edx s2)</code>	<code>(ebp edx)</code>
18: <code>(s3 &lt;- (mem ebp -4))</code>	<code>(ebp edx)</code>	<code>(ebp edx s3)</code>
19: <code>(s3 += edx)</code>	<code>(ebp edx s3)</code>	<code>(ebp edi s3)</code>
20: <code>((mem ebp -4) &lt;- s3)</code>	<code>(ebp edi s3)</code>	<code>(ebp edi)</code>
21: <code>(s4 &lt;- (mem ebp -4))</code>	<code>(ebp edi)</code>	<code>(ebp edi s4)</code>
22: <code>(s4 += edi)</code>	<code>(ebp edi s4)</code>	<code>(ebp esi s4)</code>
23: <code>((mem ebp -4) &lt;- s4)</code>	<code>(ebp esi s4)</code>	<code>(ebp esi)</code>
24: <code>(s5 &lt;- (mem ebp -4))</code>	<code>(ebp esi)</code>	<code>(ebp esi s5)</code>
25: <code>(s5 += esi)</code>	<code>(ebp esi s5)</code>	<code>(eax ebp s5)</code>
26: <code>((mem ebp -4) &lt;- s5)</code>	<code>(eax ebp s5)</code>	<code>(eax ebp)</code>
27: <code>(s6 &lt;- (mem ebp -4))</code>	<code>(eax ebp)</code>	<code>(eax ebp s6)</code>
28: <code>(s6 += eax)</code>	<code>(eax ebp s6)</code>	<code>(ebp s6)</code>
29: <code>((mem ebp -4) &lt;- s6)</code>	<code>(ebp s6)</code>	<code>()</code>



# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>()</code>
2: <code>(eax &lt;- 7)</code>	<code>()</code>	<code>()</code>
3: <code>(ebx &lt;- 7)</code>	<code>()</code>	<code>()</code>
4: <code>(ecx &lt;- 7)</code>	<code>()</code>	<code>()</code>
5: <code>(edx &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp)</code>
6: <code>(edi &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp)</code>
7: <code>(esi &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
8: <code>((mem ebp -4) &lt;- eax)</code>	<code>(eax ebp ebx)</code>	<code>(ebp ebx)</code>
9: <code>(s0 &lt;- (mem ebp -4))</code>	<code>(ebp ebx)</code>	<code>(ebp ebx s0)</code>
10: <code>(s0 += ebx)</code>	<code>(ebp ebx s0)</code>	<code>(ebp ebx s0)</code>
11: <code>((mem ebp -4) &lt;- s0)</code>	<code>(ebp ebx s0)</code>	<code>(ebp ebx)</code>
12: <code>(s1 &lt;- (mem ebp -4))</code>	<code>(ebp ebx)</code>	<code>(ebp ebx s1)</code>
13: <code>(s1 += ebx)</code>	<code>(ebp ebx ecx s1)</code>	<code>(ebp ecx s1)</code>
14: <code>((mem ebp -4) &lt;- s1)</code>	<code>(ebp ecx s1)</code>	<code>(ebp ecx)</code>
15: <code>(s2 &lt;- (mem ebp -4))</code>	<code>(ebp ecx)</code>	<code>(ebp ecx s2)</code>
16: <code>(s2 += ecx)</code>	<code>(ebp ecx edx s2)</code>	<code>(ebp edx s2)</code>
17: <code>((mem ebp -4) &lt;- s2)</code>	<code>(ebp edx s2)</code>	<code>(ebp edx)</code>
18: <code>(s3 &lt;- (mem ebp -4))</code>	<code>(ebp edx)</code>	<code>(ebp edx s3)</code>
19: <code>(s3 += edx)</code>	<code>(ebp edi edx s3)</code>	<code>(ebp edi s3)</code>
20: <code>((mem ebp -4) &lt;- s3)</code>	<code>(ebp edi s3)</code>	<code>(ebp edi)</code>
21: <code>(s4 &lt;- (mem ebp -4))</code>	<code>(ebp edi)</code>	<code>(ebp edi s4)</code>
22: <code>(s4 += edi)</code>	<code>(ebp edi esi s4)</code>	<code>(ebp esi s4)</code>
23: <code>((mem ebp -4) &lt;- s4)</code>	<code>(ebp esi s4)</code>	<code>(ebp esi)</code>
24: <code>(s5 &lt;- (mem ebp -4))</code>	<code>(ebp esi)</code>	<code>(ebp esi s5)</code>
25: <code>(s5 += esi)</code>	<code>(eax ebp esi s5)</code>	<code>(eax ebp s5)</code>
26: <code>((mem ebp -4) &lt;- s5)</code>	<code>(eax ebp s5)</code>	<code>(eax ebp)</code>
27: <code>(s6 &lt;- (mem ebp -4))</code>	<code>(eax ebp)</code>	<code>(eax ebp s6)</code>
28: <code>(s6 += eax)</code>	<code>(eax ebp s6)</code>	<code>(ebp s6)</code>
29: <code>((mem ebp -4) &lt;- s6)</code>	<code>(ebp s6)</code>	<code>()</code>

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>()</code>
2: <code>(eax &lt;- 7)</code>	<code>()</code>	<code>()</code>
3: <code>(ebx &lt;- 7)</code>	<code>()</code>	<code>()</code>
4: <code>(ecx &lt;- 7)</code>	<code>()</code>	<code>(eax ebp)</code>
5: <code>(edx &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp)</code>
6: <code>(edi &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp ebx)</code>
7: <code>(esi &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
8: <code>((mem ebp -4) &lt;- eax)</code>	<code>(eax ebp ebx)</code>	<code>(ebp ebx)</code>
9: <code>(s0 &lt;- (mem ebp -4))</code>	<code>(ebp ebx)</code>	<code>(ebp ebx s0)</code>
10: <code>(s0 += ebx)</code>	<code>(ebp ebx s0)</code>	<code>(ebp ebx s0)</code>
11: <code>((mem ebp -4) &lt;- s0)</code>	<code>(ebp ebx s0)</code>	<code>(ebp ebx)</code>
12: <code>(s1 &lt;- (mem ebp -4))</code>	<code>(ebp ebx)</code>	<code>(ebp ebx ecx s1)</code>
13: <code>(s1 += ebx)</code>	<code>(ebp ebx ecx s1)</code>	<code>(ebp ecx s1)</code>
14: <code>((mem ebp -4) &lt;- s1)</code>	<code>(ebp ecx s1)</code>	<code>(ebp ecx)</code>
15: <code>(s2 &lt;- (mem ebp -4))</code>	<code>(ebp ecx)</code>	<code>(ebp ecx edx s2)</code>
16: <code>(s2 += ecx)</code>	<code>(ebp ecx edx s2)</code>	<code>(ebp edx s2)</code>
17: <code>((mem ebp -4) &lt;- s2)</code>	<code>(ebp edx s2)</code>	<code>(ebp edx)</code>
18: <code>(s3 &lt;- (mem ebp -4))</code>	<code>(ebp edx)</code>	<code>(ebp edi edx s3)</code>
19: <code>(s3 += edx)</code>	<code>(ebp edi edx s3)</code>	<code>(ebp edi s3)</code>
20: <code>((mem ebp -4) &lt;- s3)</code>	<code>(ebp edi s3)</code>	<code>(ebp edi)</code>
21: <code>(s4 &lt;- (mem ebp -4))</code>	<code>(ebp edi)</code>	<code>(ebp edi esi s4)</code>
22: <code>(s4 += edi)</code>	<code>(ebp edi esi s4)</code>	<code>(ebp esi s4)</code>
23: <code>((mem ebp -4) &lt;- s4)</code>	<code>(ebp esi s4)</code>	<code>(ebp esi)</code>
24: <code>(s5 &lt;- (mem ebp -4))</code>	<code>(ebp esi)</code>	<code>(eax ebp esi s5)</code>
25: <code>(s5 += esi)</code>	<code>(eax ebp esi s5)</code>	<code>(eax ebp s5)</code>
26: <code>((mem ebp -4) &lt;- s5)</code>	<code>(eax ebp s5)</code>	<code>(eax ebp)</code>
27: <code>(s6 &lt;- (mem ebp -4))</code>	<code>(eax ebp)</code>	<code>(eax ebp s6)</code>
28: <code>(s6 += eax)</code>	<code>(eax ebp s6)</code>	<code>(ebp s6)</code>
29: <code>((mem ebp -4) &lt;- s6)</code>	<code>(ebp s6)</code>	<code>()</code>

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	()	()
2: <code>(eax &lt;- 7)</code>	()	()
3: <code>(ebx &lt;- 7)</code>	()	()
4: <code>(ecx &lt;- 7)</code>	(eax ebp)	(eax ebp)
5: <code>(edx &lt;- 7)</code>	(eax ebp)	(eax ebp)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx)	(ebp ebx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx)	(ebp ebx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx s0)	(ebp ebx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx s0)	(ebp ebx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx)	(ebp ebx ecx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx s1)	(ebp ecx s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx s1)	(ebp ecx)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edx)	(ebp ecx edx s2)
16: <code>(s2 += ecx)</code>	(ebp ecx edx s2)	(ebp edx s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp edx s2)	(ebp edx)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(ebp edi edx)	(ebp edi edx s3)
19: <code>(s3 += edx)</code>	(ebp edi edx s3)	(ebp edi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(ebp edi s3)	(ebp edi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(ebp edi esi)	(ebp edi esi s4)
22: <code>(s4 += edi)</code>	(ebp edi esi s4)	(ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(ebp esi s4)	(ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>()</code>
2: <code>(eax &lt;- 7)</code>	<code>()</code>	<code>()</code>
3: <code>(ebx &lt;- 7)</code>	<code>()</code>	<code>(eax ebp)</code>
4: <code>(ecx &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp)</code>
5: <code>(edx &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp ebx)</code>
6: <code>(edi &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
7: <code>(esi &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
8: <code>((mem ebp -4) &lt;- eax)</code>	<code>(eax ebp ebx)</code>	<code>(ebp ebx)</code>
9: <code>(s0 &lt;- (mem ebp -4))</code>	<code>(ebp ebx)</code>	<code>(ebp ebx s0)</code>
10: <code>(s0 += ebx)</code>	<code>(ebp ebx s0)</code>	<code>(ebp ebx s0)</code>
11: <code>((mem ebp -4) &lt;- s0)</code>	<code>(ebp ebx s0)</code>	<code>(ebp ebx ecx)</code>
12: <code>(s1 &lt;- (mem ebp -4))</code>	<code>(ebp ebx ecx)</code>	<code>(ebp ebx ecx s1)</code>
13: <code>(s1 += ebx)</code>	<code>(ebp ebx ecx s1)</code>	<code>(ebp ecx s1)</code>
14: <code>((mem ebp -4) &lt;- s1)</code>	<code>(ebp ecx s1)</code>	<code>(ebp ecx edx)</code>
15: <code>(s2 &lt;- (mem ebp -4))</code>	<code>(ebp ecx edx)</code>	<code>(ebp ecx edx s2)</code>
16: <code>(s2 += ecx)</code>	<code>(ebp ecx edx s2)</code>	<code>(ebp edx s2)</code>
17: <code>((mem ebp -4) &lt;- s2)</code>	<code>(ebp edx s2)</code>	<code>(ebp edi edx)</code>
18: <code>(s3 &lt;- (mem ebp -4))</code>	<code>(ebp edi edx)</code>	<code>(ebp edi edx s3)</code>
19: <code>(s3 += edx)</code>	<code>(ebp edi edx s3)</code>	<code>(ebp edi s3)</code>
20: <code>((mem ebp -4) &lt;- s3)</code>	<code>(ebp edi s3)</code>	<code>(ebp edi esi)</code>
21: <code>(s4 &lt;- (mem ebp -4))</code>	<code>(ebp edi esi)</code>	<code>(ebp edi esi s4)</code>
22: <code>(s4 += edi)</code>	<code>(ebp edi esi s4)</code>	<code>(ebp esi s4)</code>
23: <code>((mem ebp -4) &lt;- s4)</code>	<code>(ebp esi s4)</code>	<code>(eax ebp esi)</code>
24: <code>(s5 &lt;- (mem ebp -4))</code>	<code>(eax ebp esi)</code>	<code>(eax ebp esi s5)</code>
25: <code>(s5 += esi)</code>	<code>(eax ebp esi s5)</code>	<code>(eax ebp s5)</code>
26: <code>((mem ebp -4) &lt;- s5)</code>	<code>(eax ebp s5)</code>	<code>(eax ebp)</code>
27: <code>(s6 &lt;- (mem ebp -4))</code>	<code>(eax ebp)</code>	<code>(eax ebp s6)</code>
28: <code>(s6 += eax)</code>	<code>(eax ebp s6)</code>	<code>(ebp s6)</code>
29: <code>((mem ebp -4) &lt;- s6)</code>	<code>(ebp s6)</code>	<code>()</code>

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>()</code>
2: <code>(eax &lt;- 7)</code>	<code>()</code>	<code>()</code>
3: <code>(ebx &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp)</code>
4: <code>(ecx &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp)</code>
5: <code>(edx &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
6: <code>(edi &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
7: <code>(esi &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
8: <code>((mem ebp -4) &lt;- eax)</code>	<code>(eax ebp ebx)</code>	<code>(ebp ebx)</code>
9: <code>(s0 &lt;- (mem ebp -4))</code>	<code>(ebp ebx)</code>	<code>(ebp ebx s0)</code>
10: <code>(s0 += ebx)</code>	<code>(ebp ebx s0)</code>	<code>(ebp ebx s0)</code>
11: <code>((mem ebp -4) &lt;- s0)</code>	<code>(ebp ebx ecx s0)</code>	<code>(ebp ebx ecx)</code>
12: <code>(s1 &lt;- (mem ebp -4))</code>	<code>(ebp ebx ecx)</code>	<code>(ebp ebx ecx s1)</code>
13: <code>(s1 += ebx)</code>	<code>(ebp ebx ecx s1)</code>	<code>(ebp ecx s1)</code>
14: <code>((mem ebp -4) &lt;- s1)</code>	<code>(ebp ecx edx s1)</code>	<code>(ebp ecx edx)</code>
15: <code>(s2 &lt;- (mem ebp -4))</code>	<code>(ebp ecx edx)</code>	<code>(ebp ecx edx s2)</code>
16: <code>(s2 += ecx)</code>	<code>(ebp ecx edx s2)</code>	<code>(ebp edx s2)</code>
17: <code>((mem ebp -4) &lt;- s2)</code>	<code>(ebp edi edx s2)</code>	<code>(ebp edi edx)</code>
18: <code>(s3 &lt;- (mem ebp -4))</code>	<code>(ebp edi edx)</code>	<code>(ebp edi edx s3)</code>
19: <code>(s3 += edx)</code>	<code>(ebp edi edx s3)</code>	<code>(ebp edi s3)</code>
20: <code>((mem ebp -4) &lt;- s3)</code>	<code>(ebp edi esi s3)</code>	<code>(ebp edi esi)</code>
21: <code>(s4 &lt;- (mem ebp -4))</code>	<code>(ebp edi esi)</code>	<code>(ebp edi esi s4)</code>
22: <code>(s4 += edi)</code>	<code>(ebp edi esi s4)</code>	<code>(ebp esi s4)</code>
23: <code>((mem ebp -4) &lt;- s4)</code>	<code>(eax ebp esi s4)</code>	<code>(eax ebp esi)</code>
24: <code>(s5 &lt;- (mem ebp -4))</code>	<code>(eax ebp esi)</code>	<code>(eax ebp esi s5)</code>
25: <code>(s5 += esi)</code>	<code>(eax ebp esi s5)</code>	<code>(eax ebp s5)</code>
26: <code>((mem ebp -4) &lt;- s5)</code>	<code>(eax ebp s5)</code>	<code>(eax ebp)</code>
27: <code>(s6 &lt;- (mem ebp -4))</code>	<code>(eax ebp)</code>	<code>(eax ebp s6)</code>
28: <code>(s6 += eax)</code>	<code>(eax ebp s6)</code>	<code>(ebp s6)</code>
29: <code>((mem ebp -4) &lt;- s6)</code>	<code>(ebp s6)</code>	<code>()</code>

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>()</code>
2: <code>(eax &lt;- 7)</code>	<code>()</code>	<code>(eax ebp)</code>
3: <code>(ebx &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp)</code>
4: <code>(ecx &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp ebx)</code>
5: <code>(edx &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
6: <code>(edi &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
7: <code>(esi &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
8: <code>((mem ebp -4) &lt;- eax)</code>	<code>(eax ebp ebx)</code>	<code>(ebp ebx)</code>
9: <code>(s0 &lt;- (mem ebp -4))</code>	<code>(ebp ebx)</code>	<code>(ebp ebx s0)</code>
10: <code>(s0 += ebx)</code>	<code>(ebp ebx s0)</code>	<code>(ebp ebx ecx s0)</code>
11: <code>((mem ebp -4) &lt;- s0)</code>	<code>(ebp ebx ecx s0)</code>	<code>(ebp ebx ecx)</code>
12: <code>(s1 &lt;- (mem ebp -4))</code>	<code>(ebp ebx ecx)</code>	<code>(ebp ebx ecx s1)</code>
13: <code>(s1 += ebx)</code>	<code>(ebp ebx ecx s1)</code>	<code>(ebp ecx edx s1)</code>
14: <code>((mem ebp -4) &lt;- s1)</code>	<code>(ebp ecx edx s1)</code>	<code>(ebp ecx edx)</code>
15: <code>(s2 &lt;- (mem ebp -4))</code>	<code>(ebp ecx edx)</code>	<code>(ebp ecx edx s2)</code>
16: <code>(s2 += ecx)</code>	<code>(ebp ecx edx s2)</code>	<code>(ebp edi edx s2)</code>
17: <code>((mem ebp -4) &lt;- s2)</code>	<code>(ebp edi edx s2)</code>	<code>(ebp edi edx)</code>
18: <code>(s3 &lt;- (mem ebp -4))</code>	<code>(ebp edi edx)</code>	<code>(ebp edi edx s3)</code>
19: <code>(s3 += edx)</code>	<code>(ebp edi edx s3)</code>	<code>(ebp edi esi s3)</code>
20: <code>((mem ebp -4) &lt;- s3)</code>	<code>(ebp edi esi s3)</code>	<code>(ebp edi esi)</code>
21: <code>(s4 &lt;- (mem ebp -4))</code>	<code>(ebp edi esi)</code>	<code>(ebp edi esi s4)</code>
22: <code>(s4 += edi)</code>	<code>(ebp edi esi s4)</code>	<code>(eax ebp esi s4)</code>
23: <code>((mem ebp -4) &lt;- s4)</code>	<code>(eax ebp esi s4)</code>	<code>(eax ebp esi)</code>
24: <code>(s5 &lt;- (mem ebp -4))</code>	<code>(eax ebp esi)</code>	<code>(eax ebp esi s5)</code>
25: <code>(s5 += esi)</code>	<code>(eax ebp esi s5)</code>	<code>(eax ebp s5)</code>
26: <code>((mem ebp -4) &lt;- s5)</code>	<code>(eax ebp s5)</code>	<code>(eax ebp)</code>
27: <code>(s6 &lt;- (mem ebp -4))</code>	<code>(eax ebp)</code>	<code>(eax ebp s6)</code>
28: <code>(s6 += eax)</code>	<code>(eax ebp s6)</code>	<code>(ebp s6)</code>
29: <code>((mem ebp -4) &lt;- s6)</code>	<code>(ebp s6)</code>	<code>()</code>

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>()</code>
2: <code>(eax &lt;- 7)</code>	<code>(ebp)</code>	<code>(eax ebp)</code>
3: <code>(ebx &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp)</code>
4: <code>(ecx &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
5: <code>(edx &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
6: <code>(edi &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
7: <code>(esi &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
8: <code>((mem ebp -4) &lt;- eax)</code>	<code>(eax ebp ebx)</code>	<code>(ebp ebx)</code>
9: <code>(s0 &lt;- (mem ebp -4))</code>	<code>(ebp ebx)</code>	<code>(ebp ebx s0)</code>
10: <code>(s0 += ebx)</code>	<code>(ebp ebx ecx s0)</code>	<code>(ebp ebx ecx s0)</code>
11: <code>((mem ebp -4) &lt;- s0)</code>	<code>(ebp ebx ecx s0)</code>	<code>(ebp ebx ecx)</code>
12: <code>(s1 &lt;- (mem ebp -4))</code>	<code>(ebp ebx ecx)</code>	<code>(ebp ebx ecx s1)</code>
13: <code>(s1 += ebx)</code>	<code>(ebp ebx ecx edx s1)</code>	<code>(ebp ecx edx s1)</code>
14: <code>((mem ebp -4) &lt;- s1)</code>	<code>(ebp ecx edx s1)</code>	<code>(ebp ecx edx)</code>
15: <code>(s2 &lt;- (mem ebp -4))</code>	<code>(ebp ecx edx)</code>	<code>(ebp ecx edx s2)</code>
16: <code>(s2 += ecx)</code>	<code>(ebp ecx edi edx s2)</code>	<code>(ebp edi edx s2)</code>
17: <code>((mem ebp -4) &lt;- s2)</code>	<code>(ebp edi edx s2)</code>	<code>(ebp edi edx)</code>
18: <code>(s3 &lt;- (mem ebp -4))</code>	<code>(ebp edi edx)</code>	<code>(ebp edi edx s3)</code>
19: <code>(s3 += edx)</code>	<code>(ebp edi edx esi s3)</code>	<code>(ebp edi esi s3)</code>
20: <code>((mem ebp -4) &lt;- s3)</code>	<code>(ebp edi esi s3)</code>	<code>(ebp edi esi)</code>
21: <code>(s4 &lt;- (mem ebp -4))</code>	<code>(ebp edi esi)</code>	<code>(ebp edi esi s4)</code>
22: <code>(s4 += edi)</code>	<code>(eax ebp edi esi s4)</code>	<code>(eax ebp esi s4)</code>
23: <code>((mem ebp -4) &lt;- s4)</code>	<code>(eax ebp esi s4)</code>	<code>(eax ebp esi)</code>
24: <code>(s5 &lt;- (mem ebp -4))</code>	<code>(eax ebp esi)</code>	<code>(eax ebp esi s5)</code>
25: <code>(s5 += esi)</code>	<code>(eax ebp esi s5)</code>	<code>(eax ebp s5)</code>
26: <code>((mem ebp -4) &lt;- s5)</code>	<code>(eax ebp s5)</code>	<code>(eax ebp)</code>
27: <code>(s6 &lt;- (mem ebp -4))</code>	<code>(eax ebp)</code>	<code>(eax ebp s6)</code>
28: <code>(s6 += eax)</code>	<code>(eax ebp s6)</code>	<code>(ebp s6)</code>
29: <code>((mem ebp -4) &lt;- s6)</code>	<code>(ebp s6)</code>	<code>()</code>

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	<code>()</code>	<code>(ebp)</code>
2: <code>(eax &lt;- 7)</code>	<code>(ebp)</code>	<code>(eax ebp)</code>
3: <code>(ebx &lt;- 7)</code>	<code>(eax ebp)</code>	<code>(eax ebp ebx)</code>
4: <code>(ecx &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
5: <code>(edx &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
6: <code>(edi &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
7: <code>(esi &lt;- 7)</code>	<code>(eax ebp ebx)</code>	<code>(eax ebp ebx)</code>
8: <code>((mem ebp -4) &lt;- eax)</code>	<code>(eax ebp ebx)</code>	<code>(ebp ebx)</code>
9: <code>(s0 &lt;- (mem ebp -4))</code>	<code>(ebp ebx)</code>	<code>(ebp ebx ecx s0)</code>
10: <code>(s0 += ebx)</code>	<code>(ebp ebx ecx s0)</code>	<code>(ebp ebx ecx s0)</code>
11: <code>((mem ebp -4) &lt;- s0)</code>	<code>(ebp ebx ecx s0)</code>	<code>(ebp ebx ecx)</code>
12: <code>(s1 &lt;- (mem ebp -4))</code>	<code>(ebp ebx ecx)</code>	<code>(ebp ebx ecx edx s1)</code>
13: <code>(s1 += ebx)</code>	<code>(ebp ebx ecx edx s1)</code>	<code>(ebp ecx edx s1)</code>
14: <code>((mem ebp -4) &lt;- s1)</code>	<code>(ebp ecx edx s1)</code>	<code>(ebp ecx edx)</code>
15: <code>(s2 &lt;- (mem ebp -4))</code>	<code>(ebp ecx edx)</code>	<code>(ebp ecx edi edx s2)</code>
16: <code>(s2 += ecx)</code>	<code>(ebp ecx edi edx s2)</code>	<code>(ebp edi edx s2)</code>
17: <code>((mem ebp -4) &lt;- s2)</code>	<code>(ebp edi edx s2)</code>	<code>(ebp edi edx)</code>
18: <code>(s3 &lt;- (mem ebp -4))</code>	<code>(ebp edi edx)</code>	<code>(ebp edi edx esi s3)</code>
19: <code>(s3 += edx)</code>	<code>(ebp edi edx esi s3)</code>	<code>(ebp edi esi s3)</code>
20: <code>((mem ebp -4) &lt;- s3)</code>	<code>(ebp edi esi s3)</code>	<code>(ebp edi esi)</code>
21: <code>(s4 &lt;- (mem ebp -4))</code>	<code>(ebp edi esi)</code>	<code>(eax ebp edi esi s4)</code>
22: <code>(s4 += edi)</code>	<code>(eax ebp edi esi s4)</code>	<code>(eax ebp esi s4)</code>
23: <code>((mem ebp -4) &lt;- s4)</code>	<code>(eax ebp esi s4)</code>	<code>(eax ebp esi)</code>
24: <code>(s5 &lt;- (mem ebp -4))</code>	<code>(eax ebp esi)</code>	<code>(eax ebp esi s5)</code>
25: <code>(s5 += esi)</code>	<code>(eax ebp esi s5)</code>	<code>(eax ebp s5)</code>
26: <code>((mem ebp -4) &lt;- s5)</code>	<code>(eax ebp s5)</code>	<code>(eax ebp)</code>
27: <code>(s6 &lt;- (mem ebp -4))</code>	<code>(eax ebp)</code>	<code>(eax ebp s6)</code>
28: <code>(s6 += eax)</code>	<code>(eax ebp s6)</code>	<code>(ebp s6)</code>
29: <code>((mem ebp -4) &lt;- s6)</code>	<code>(ebp s6)</code>	<code>()</code>



# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx)	(ebp ebx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx)	(ebp ebx ecx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx s0)	(ebp ebx ecx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx s0)	(ebp ebx ecx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edx)	(ebp ebx ecx edx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edx s1)	(ebp ecx edx s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edx s1)	(ebp ecx edx)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edi edx)	(ebp ecx edi edx s2)
16: <code>(s2 += ecx)</code>	(ebp ecx edi edx s2)	(ebp edi edx s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp edi edx s2)	(ebp edi edx)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(ebp edi edx esi)	(ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(ebp edi edx esi s3)	(ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(ebp edi esi s3)	(ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx)	(ebp ebx ecx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx)	(ebp ebx ecx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx s0)	(ebp ebx ecx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx s0)	(ebp ebx ecx edx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edx)	(ebp ebx ecx edx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edx s1)	(ebp ecx edx s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edx s1)	(ebp ecx edi edx)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edi edx)	(ebp ecx edi edx s2)
16: <code>(s2 += ecx)</code>	(ebp ecx edi edx s2)	(ebp edi edx s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp edi edx s2)	(ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(ebp edi edx esi)	(ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(ebp edi edx esi s3)	(ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx)	(ebp ebx ecx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx)	(ebp ebx ecx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx s0)	(ebp ebx ecx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edx s0)	(ebp ebx ecx edx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edx)	(ebp ebx ecx edx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edx s1)	(ebp ecx edx s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edi edx s1)	(ebp ecx edi edx)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edi edx)	(ebp ecx edi edx s2)
16: <code>(s2 += ecx)</code>	(ebp ecx edi edx s2)	(ebp edi edx s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp edi edx esi s2)	(ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(ebp edi edx esi)	(ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(ebp edi edx esi s3)	(ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx)	(ebp ebx ecx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx)	(ebp ebx ecx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx s0)	(ebp ebx ecx edx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edx s0)	(ebp ebx ecx edx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edx)	(ebp ebx ecx edx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edx s1)	(ebp ecx edi edx s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edi edx s1)	(ebp ecx edi edx)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edi edx)	(ebp ecx edi edx s2)
16: <code>(s2 += ecx)</code>	(ebp ecx edi edx s2)	(ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp edi edx esi s2)	(ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(ebp edi edx esi)	(ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx)	(ebp ebx ecx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx)	(ebp ebx ecx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edx s0)	(ebp ebx ecx edx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edx s0)	(ebp ebx ecx edx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edx)	(ebp ebx ecx edx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edi edx s1)	(ebp ecx edi edx s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edi edx s1)	(ebp ecx edi edx)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edi edx)	(ebp ecx edi edx s2)
16: <code>(s2 += ecx)</code>	(ebp ecx edi edx esi s2)	(ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp edi edx esi s2)	(ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(ebp edi edx esi)	(ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx)	(ebp ebx ecx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx)	(ebp ebx ecx edx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edx s0)	(ebp ebx ecx edx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edx s0)	(ebp ebx ecx edx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edx)	(ebp ebx ecx edi edx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edi edx s1)	(ebp ecx edi edx s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edi edx s1)	(ebp ecx edi edx)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edi edx)	(ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(ebp ecx edi edx esi s2)	(ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp edi edx esi s2)	(ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx)	(ebp ebx ecx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edx)	(ebp ebx ecx edx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edx s0)	(ebp ebx ecx edx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edx s0)	(ebp ebx ecx edx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx)	(ebp ebx ecx edi edx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edi edx s1)	(ebp ecx edi edx s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edi edx s1)	(ebp ecx edi edx)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edi edx esi)	(ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(ebp ecx edi edx esi s2)	(ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp edi edx esi s2)	(ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx)	(ebp ebx ecx edx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edx)	(ebp ebx ecx edx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edx s0)	(ebp ebx ecx edx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edx s0)	(ebp ebx ecx edi edx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx)	(ebp ebx ecx edi edx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edi edx s1)	(ebp ecx edi edx s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edi edx s1)	(ebp ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edi edx esi)	(ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(ebp ecx edi edx esi s2)	(ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(ebp edi edx esi s2)	(eax ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()



# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edx)	(ebp ebx ecx edx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edx)	(ebp ebx ecx edx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edx s0)	(ebp ebx ecx edx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edi edx s0)	(ebp ebx ecx edi edx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx)	(ebp ebx ecx edi edx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edi edx s1)	(ebp ecx edi edx s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edi edx esi s1)	(ebp ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edi edx esi)	(ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(ebp ecx edi edx esi s2)	(ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp edi edx esi s2)	(eax ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edx)	(ebp ebx ecx edx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edx)	(ebp ebx ecx edx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edx s0)	(ebp ebx ecx edi edx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edi edx s0)	(ebp ebx ecx edi edx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx)	(ebp ebx ecx edi edx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edi edx s1)	(ebp ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edi edx esi s1)	(ebp ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edi edx esi)	(ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(ebp ecx edi edx esi s2)	(eax ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp edi edx esi s2)	(eax ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edx)	(ebp ebx ecx edx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edx)	(ebp ebx ecx edx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edi edx s0)	(ebp ebx ecx edi edx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edi edx s0)	(ebp ebx ecx edi edx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx)	(ebp ebx ecx edi edx s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edi edx esi s1)	(ebp ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edi edx esi s1)	(ebp ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edi edx esi)	(ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ecx edi edx esi s2)	(eax ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp edi edx esi s2)	(eax ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edx)	(ebp ebx ecx edx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edx)	(ebp ebx ecx edi edx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edi edx s0)	(ebp ebx ecx edi edx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edi edx s0)	(ebp ebx ecx edi edx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx)	(ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edi edx esi s1)	(ebp ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edi edx esi s1)	(ebp ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(ebp ecx edi edx esi)	(eax ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ecx edi edx esi s2)	(eax ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp edi edx esi s2)	(eax ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edx)	(ebp ebx ecx edx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx)	(ebp ebx ecx edi edx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edi edx s0)	(ebp ebx ecx edi edx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edi edx s0)	(ebp ebx ecx edi edx)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edi edx esi s1)	(ebp ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edi edx esi s1)	(ebp ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ecx edi edx esi)	(eax ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ecx edi edx esi s2)	(eax ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp edi edx esi s2)	(eax ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edx)	(ebp ebx ecx edi edx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx)	(ebp ebx ecx edi edx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edi edx s0)	(ebp ebx ecx edi edx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edi edx s0)	(ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edi edx esi s1)	(ebp ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(ebp ecx edi edx esi s1)	(eax ebp ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ecx edi edx esi)	(eax ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ecx edi edx esi s2)	(eax ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp edi edx esi s2)	(eax ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edi edx)	(ebp ebx ecx edi edx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx)	(ebp ebx ecx edi edx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edi edx s0)	(ebp ebx ecx edi edx s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edi edx esi s0)	(ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edi edx esi s1)	(ebp ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(eax ebp ecx edi edx esi s1)	(eax ebp ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ecx edi edx esi)	(eax ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ecx edi edx esi s2)	(eax ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp edi edx esi s2)	(eax ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edi edx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edi edx)	(ebp ebx ecx edi edx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx)	(ebp ebx ecx edi edx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edi edx s0)	(ebp ebx ecx edi edx esi s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edi edx esi s0)	(ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(ebp ebx ecx edi edx esi s1)	(eax ebp ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(eax ebp ecx edi edx esi s1)	(eax ebp ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ecx edi edx esi)	(eax ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ecx edi edx esi s2)	(eax ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp edi edx esi s2)	(eax ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()



# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edi edx)	(eax ebp ebx ecx edi edx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edi edx)	(ebp ebx ecx edi edx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx)	(ebp ebx ecx edi edx s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edi edx esi s0)	(ebp ebx ecx edi edx esi s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edi edx esi s0)	(ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(eax ebp ecx edi edx esi s1)	(eax ebp ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ecx edi edx esi)	(eax ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ecx edi edx esi s2)	(eax ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp edi edx esi s2)	(eax ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edi edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edi edx)	(eax ebp ebx ecx edi edx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edi edx)	(ebp ebx ecx edi edx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx)	(ebp ebx ecx edi edx esi s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edi edx esi s0)	(ebp ebx ecx edi edx esi s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edi edx esi s0)	(ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(eax ebp ecx edi edx esi s1)	(eax ebp ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ecx edi edx esi)	(eax ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ecx edi edx esi s2)	(eax ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp edi edx esi s2)	(eax ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edi edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edi edx)	(eax ebp ebx ecx edi edx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edi edx)	(ebp ebx ecx edi edx)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edi edx esi s0)	(ebp ebx ecx edi edx esi s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edi edx esi s0)	(ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi esi)	(eax ebp ebx ecx edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp ebx ecx edi esi s4)	(eax ebp ebx ecx edi esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp ebx ecx edi esi s4)	(eax ebp ebx ecx esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx esi)	(eax ebp ebx ecx esi s5)
25: <code>(s5 += esi)</code>	(eax ebp ebx ecx esi s5)	(eax ebp ebx ecx s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp ebx ecx s5)	(eax ebp ebx)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp ebx)	(eax ebp ebx s6)
28: <code>(s6 += eax)</code>	(eax ebp ebx s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edi edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edi edx)	(eax ebp ebx ecx edi edx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edi edx)	(ebp ebx ecx edi edx esi)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edi edx esi s0)	(ebp ebx ecx edi edx esi s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(ebp ebx ecx edi edx esi s0)	(eax ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi s4)
22: <code>(s4 += edi)</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi s5)
25: <code>(s5 += esi)</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi s6)
28: <code>(s6 += eax)</code>	(eax ebp ebx ecx edi edx esi s6)	(ebp ebx ecx edi edx esi s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp ebx ecx edi edx esi s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edi edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edi edx)	(eax ebp ebx ecx edi edx)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edi edx esi s0)	(ebp ebx ecx edi edx esi s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(eax ebp ebx ecx edi edx esi s0)	(eax ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(eax ebp ecx edi edx esi s1)	(eax ebp ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ecx edi edx esi)	(eax ebp ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ecx edi edx esi s2)	(eax ebp edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp edi edx esi s2)	(eax ebp edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp edi edx esi)	(eax ebp edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp edi edx esi s3)	(eax ebp edi esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp edi esi s3)	(eax ebp edi esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp edi esi)	(eax ebp edi esi s4)
22: <code>(s4 += edi)</code>	(eax ebp edi esi s4)	(eax ebp esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp esi s4)	(eax ebp esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp esi)	(eax ebp esi s5)
25: <code>(s5 += esi)</code>	(eax ebp esi s5)	(eax ebp s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp s5)	(eax ebp)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp)	(eax ebp s6)
28: <code>(s6 += eax)</code>	(eax ebp s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edi edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edi edx)	(eax ebp ebx ecx edi edx esi)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi s0)
10: <code>(s0 += ebx)</code>	(ebp ebx ecx edi edx esi s0)	(eax ebp ebx ecx edi edx esi s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(eax ebp ebx ecx edi edx esi s0)	(eax ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi s4)
22: <code>(s4 += edi)</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi s5)
25: <code>(s5 += esi)</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi s6)
28: <code>(s6 += eax)</code>	(eax ebp ebx ecx edi edx esi s6)	(ebp ebx ecx edi edx esi s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp ebx ecx edi edx esi s6)	()

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edi edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edi edx)	(eax ebp ebx ecx edi edx esi)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi s0)
10: <code>(s0 += ebx)</code>	(eax ebp ebx ecx edi edx esi s0)	(eax ebp ebx ecx edi edx esi s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s4)
22: <code>(s4 += edi)</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s5)
25: <code>(s5 += esi)</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s6)
28: <code>(s6 += eax)</code>	(eax ebp ebx ecx edi edx esi s6)	(eax ebp ebx ecx edi edx esi s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp)	(ebp)

# Spilled Liveness

	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edi edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edi edx)	(eax ebp ebx ecx edi edx esi)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s0)
10: <code>(s0 += ebx)</code>	(eax ebp ebx ecx edi edx esi s0)	(eax ebp ebx ecx edi edx esi s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(eax ebp ebx ecx edi edx esi s0)	(eax ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi s4)
22: <code>(s4 += edi)</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi s5)
25: <code>(s5 += esi)</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi s6)
28: <code>(s6 += eax)</code>	(eax ebp ebx ecx edi edx esi s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

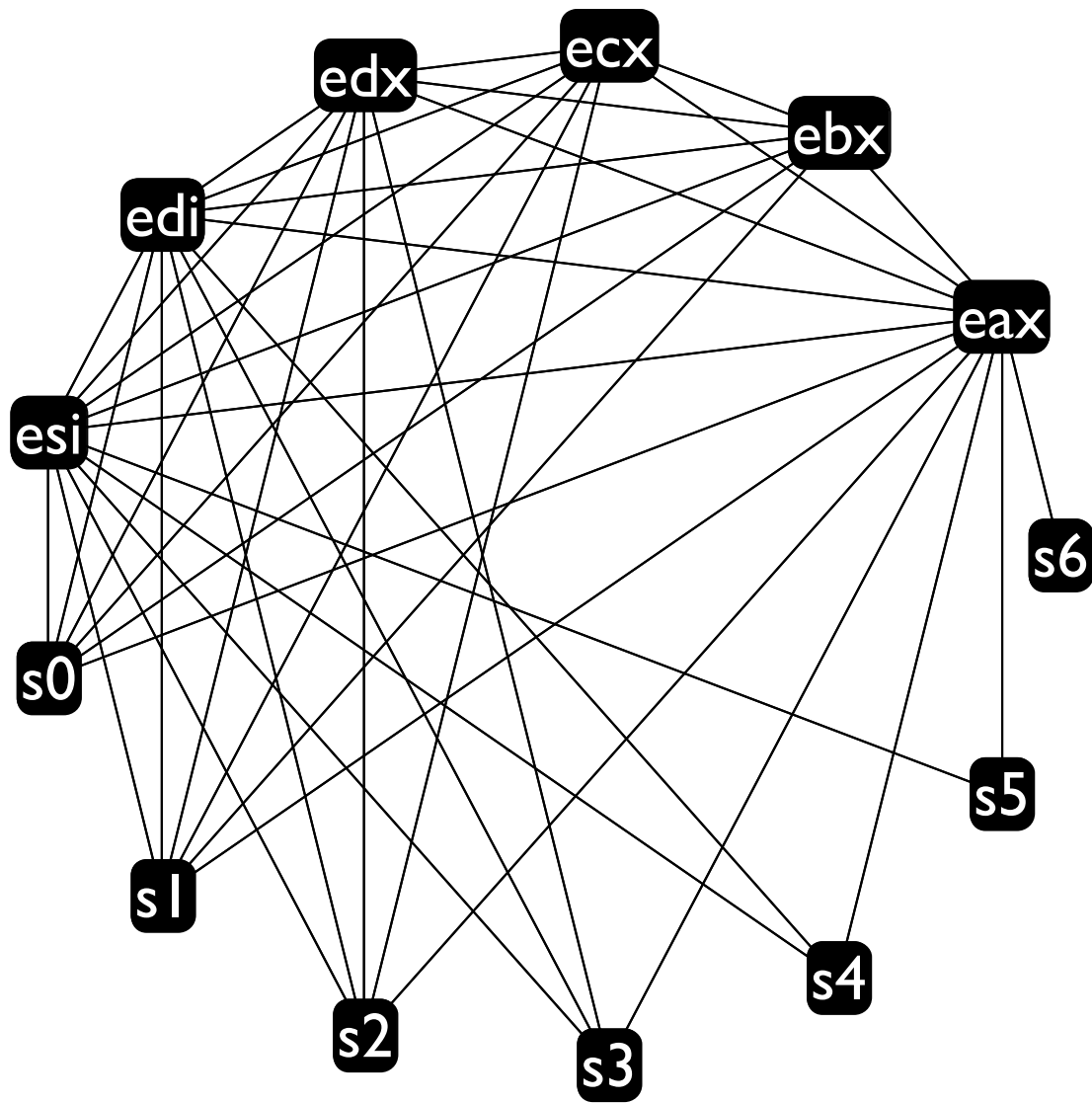


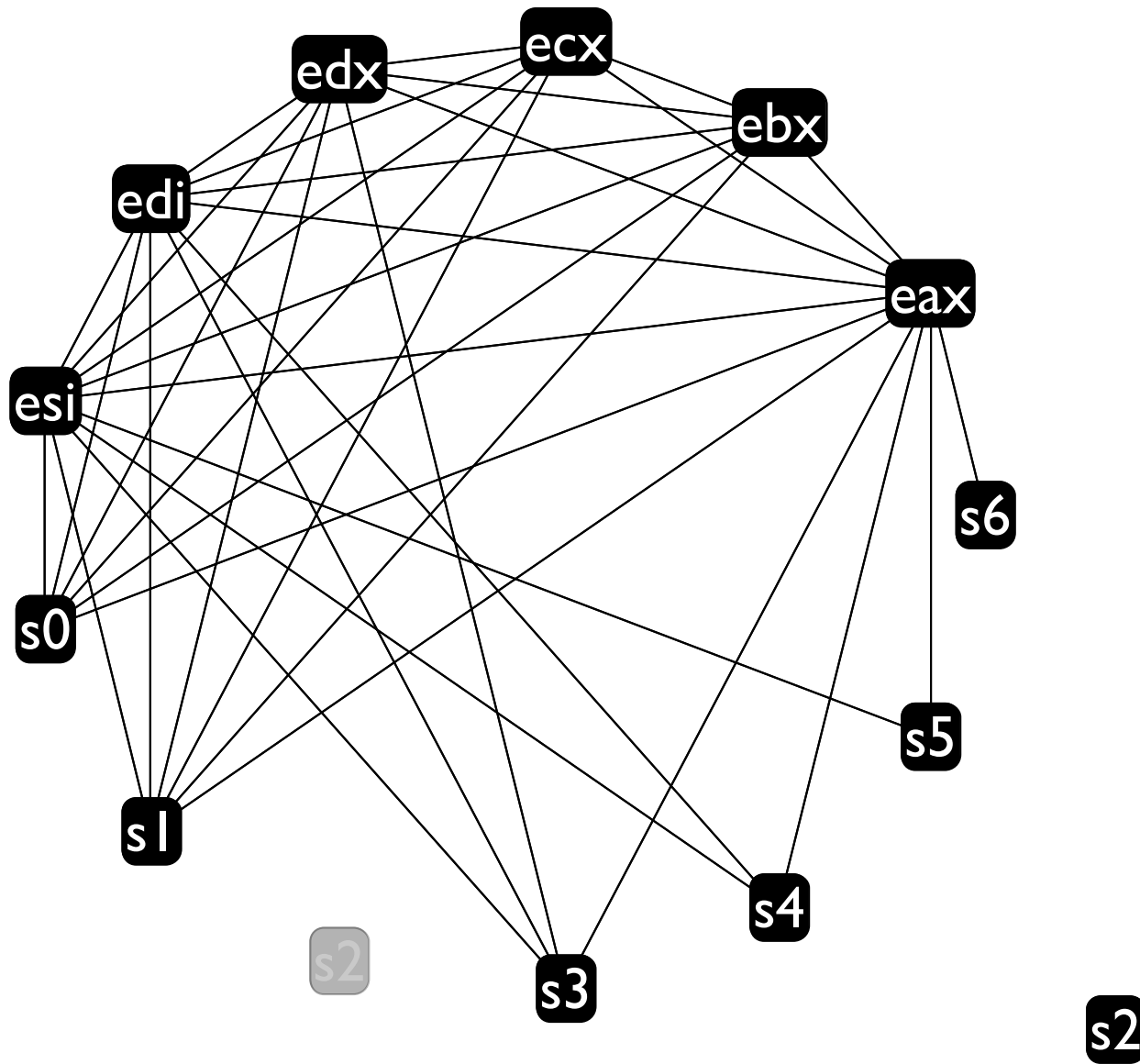
# Spilled Liveness

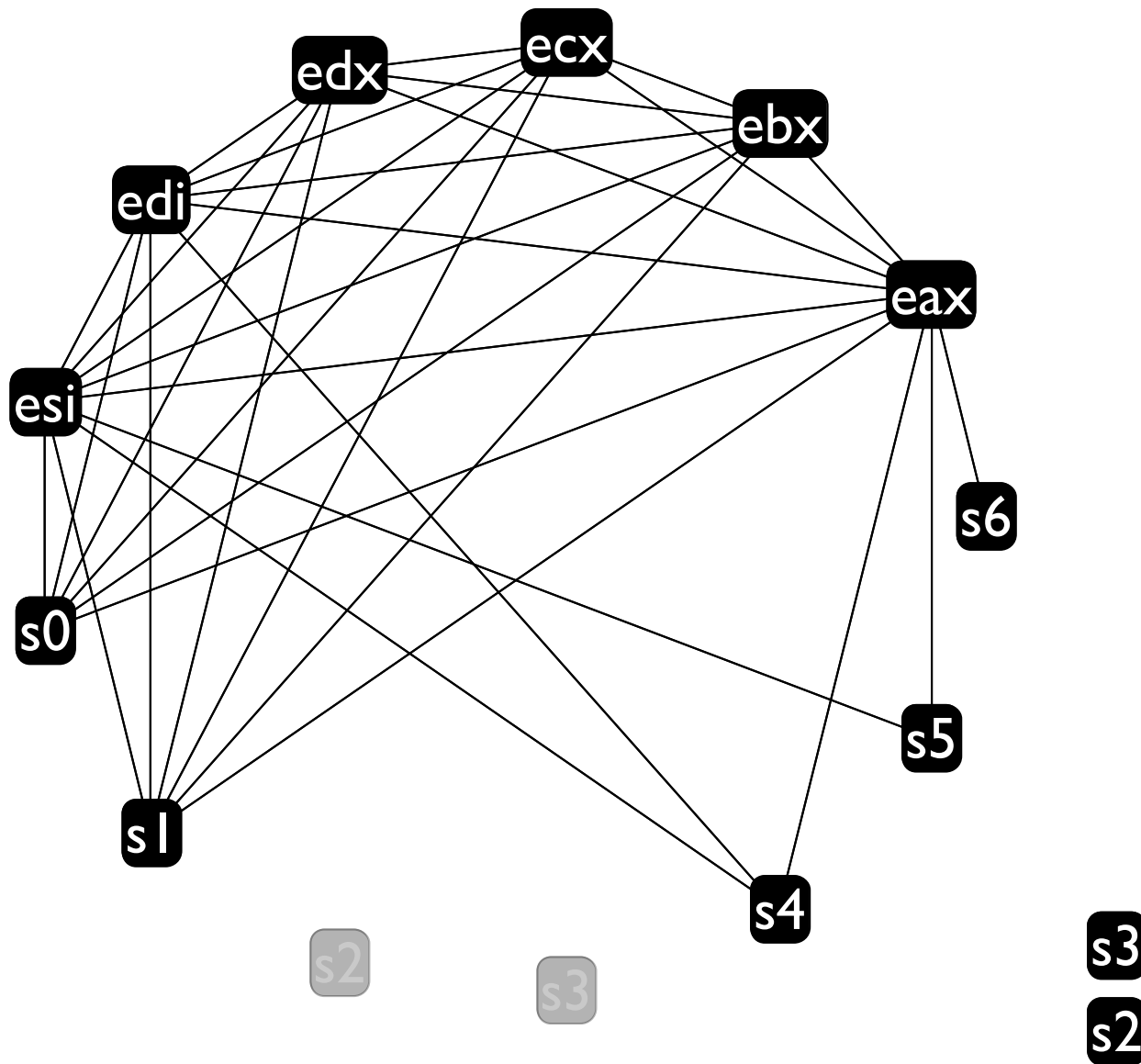
	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edi edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edi edx)	(eax ebp ebx ecx edi edx esi)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edi edx esi)	(ebp ebx ecx edi edx esi)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s0)
10: <code>(s0 += ebx)</code>	(eax ebp ebx ecx edi edx esi s0)	(eax ebp ebx ecx edi edx esi s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(eax ebp ebx ecx edi edx esi s0)	(eax ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi s4)
22: <code>(s4 += edi)</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi s5)
25: <code>(s5 += esi)</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi s6)
28: <code>(s6 += eax)</code>	(eax ebp ebx ecx edi edx esi s6)	(ebp s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(ebp s6)	()

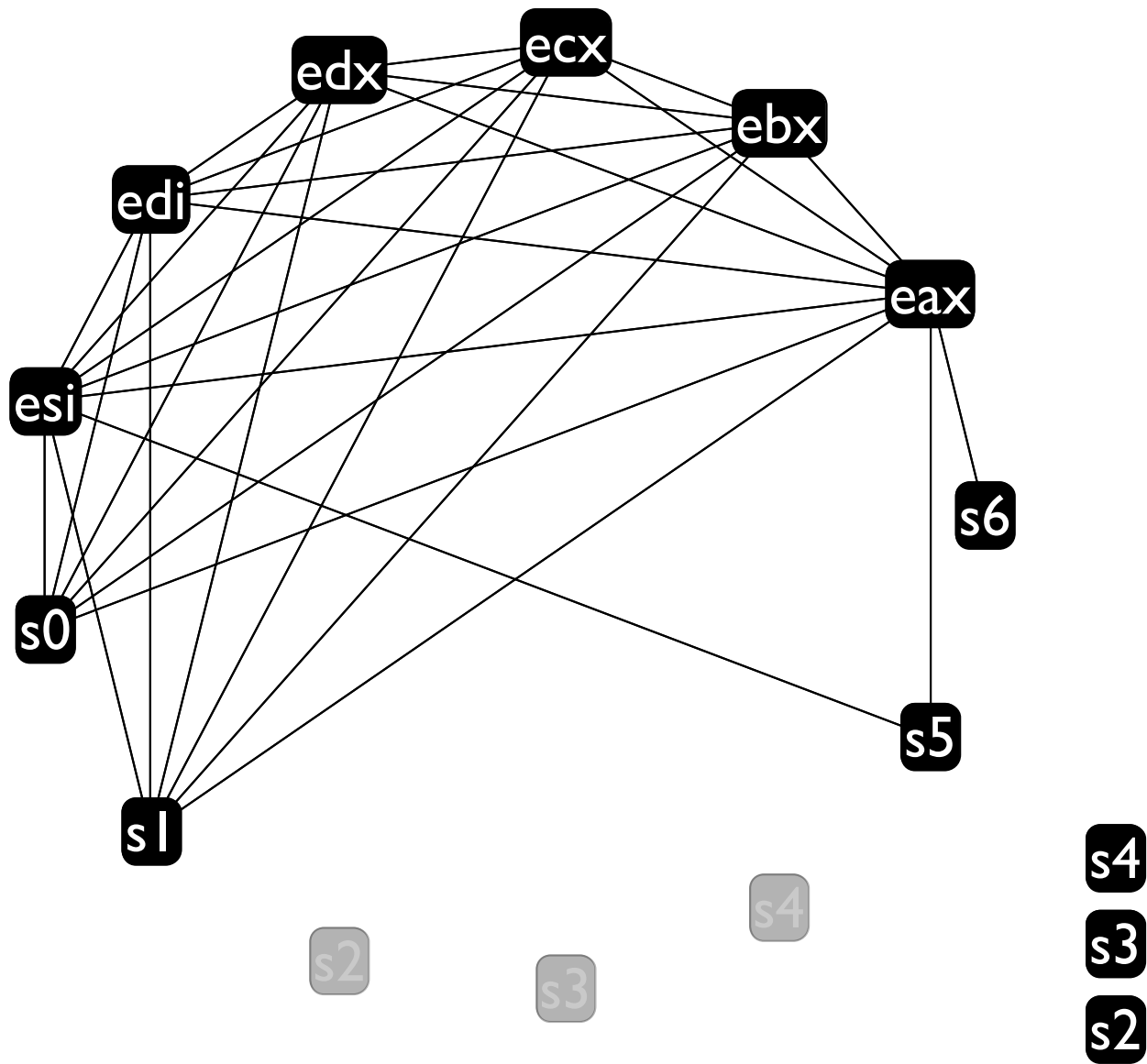
# Spilled Liveness

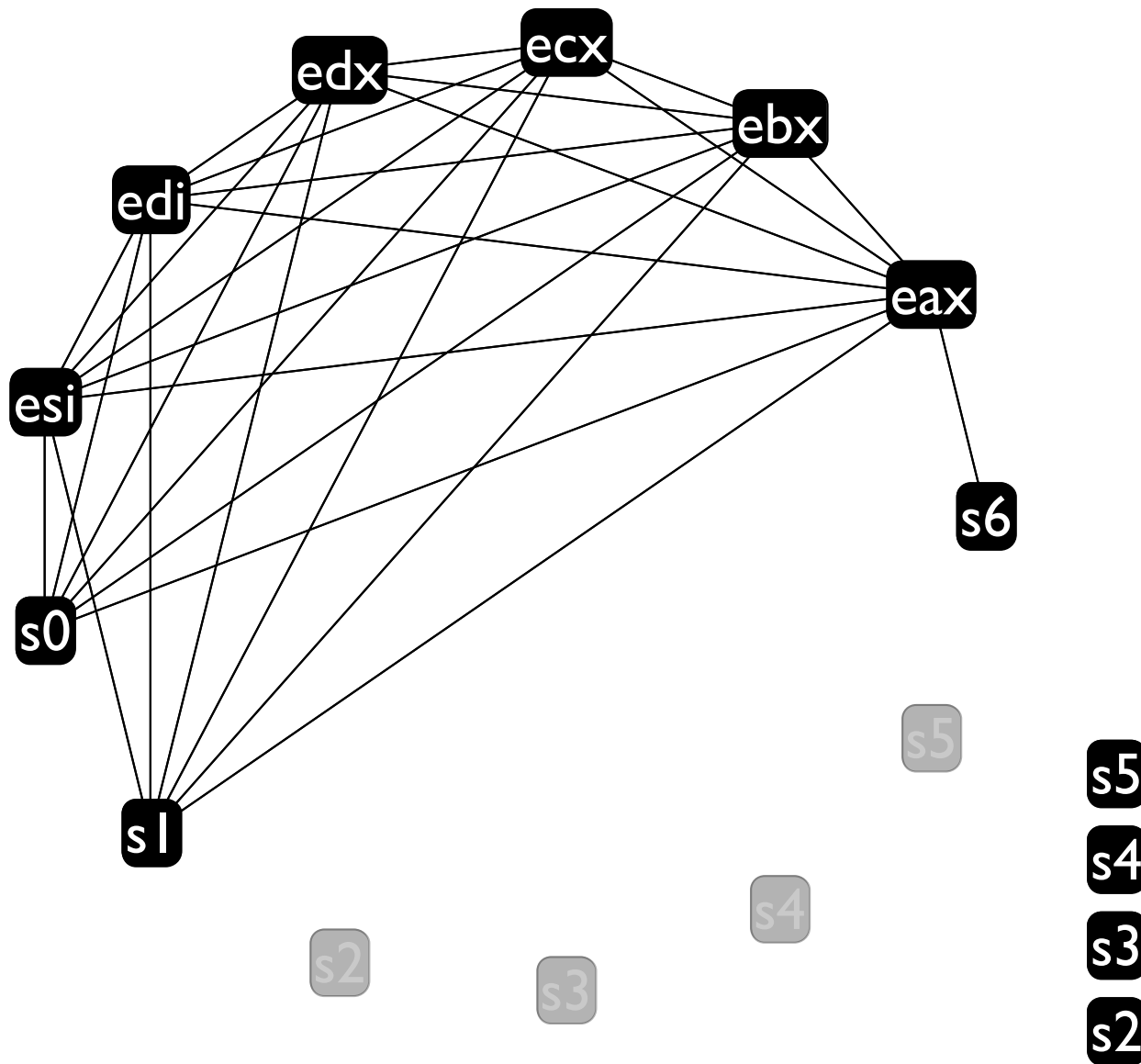
	<b>in</b>	<b>out</b>
1: <code>:imp</code>	(ebp)	(ebp)
2: <code>(eax &lt;- 7)</code>	(ebp)	(eax ebp)
3: <code>(ebx &lt;- 7)</code>	(eax ebp)	(eax ebp ebx)
4: <code>(ecx &lt;- 7)</code>	(eax ebp ebx)	(eax ebp ebx ecx)
5: <code>(edx &lt;- 7)</code>	(eax ebp ebx ecx)	(eax ebp ebx ecx edx)
6: <code>(edi &lt;- 7)</code>	(eax ebp ebx ecx edx)	(eax ebp ebx ecx edi edx)
7: <code>(esi &lt;- 7)</code>	(eax ebp ebx ecx edi edx)	(eax ebp ebx ecx edi edx esi)
8: <code>((mem ebp -4) &lt;- eax)</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi)
9: <code>(s0 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s0)
10: <code>(s0 += ebx)</code>	(eax ebp ebx ecx edi edx esi s0)	(eax ebp ebx ecx edi edx esi s0)
11: <code>((mem ebp -4) &lt;- s0)</code>	(eax ebp ebx ecx edi edx esi s0)	(eax ebp ebx ecx edi edx esi)
12: <code>(s1 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s1)
13: <code>(s1 += ebx)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi s1)
14: <code>((mem ebp -4) &lt;- s1)</code>	(eax ebp ebx ecx edi edx esi s1)	(eax ebp ebx ecx edi edx esi)
15: <code>(s2 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi)	(eax ebp ebx ecx edi edx esi s2)
16: <code>(s2 += ecx)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s2)
17: <code>((mem ebp -4) &lt;- s2)</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi)
18: <code>(s3 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s2)	(eax ebp ebx ecx edi edx esi s3)
19: <code>(s3 += edx)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi s3)
20: <code>((mem ebp -4) &lt;- s3)</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi)
21: <code>(s4 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s3)	(eax ebp ebx ecx edi edx esi s4)
22: <code>(s4 += edi)</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi s4)
23: <code>((mem ebp -4) &lt;- s4)</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi)
24: <code>(s5 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s4)	(eax ebp ebx ecx edi edx esi s5)
25: <code>(s5 += esi)</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi s5)
26: <code>((mem ebp -4) &lt;- s5)</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi)
27: <code>(s6 &lt;- (mem ebp -4))</code>	(eax ebp ebx ecx edi edx esi s5)	(eax ebp ebx ecx edi edx esi s6)
28: <code>(s6 += eax)</code>	(eax ebp ebx ecx edi edx esi s6)	(eax ebp ebx ecx edi edx esi s6)
29: <code>((mem ebp -4) &lt;- s6)</code>	(eax ebp ebx ecx edi edx esi s6)	(eax ebp ebx ecx edi edx esi)

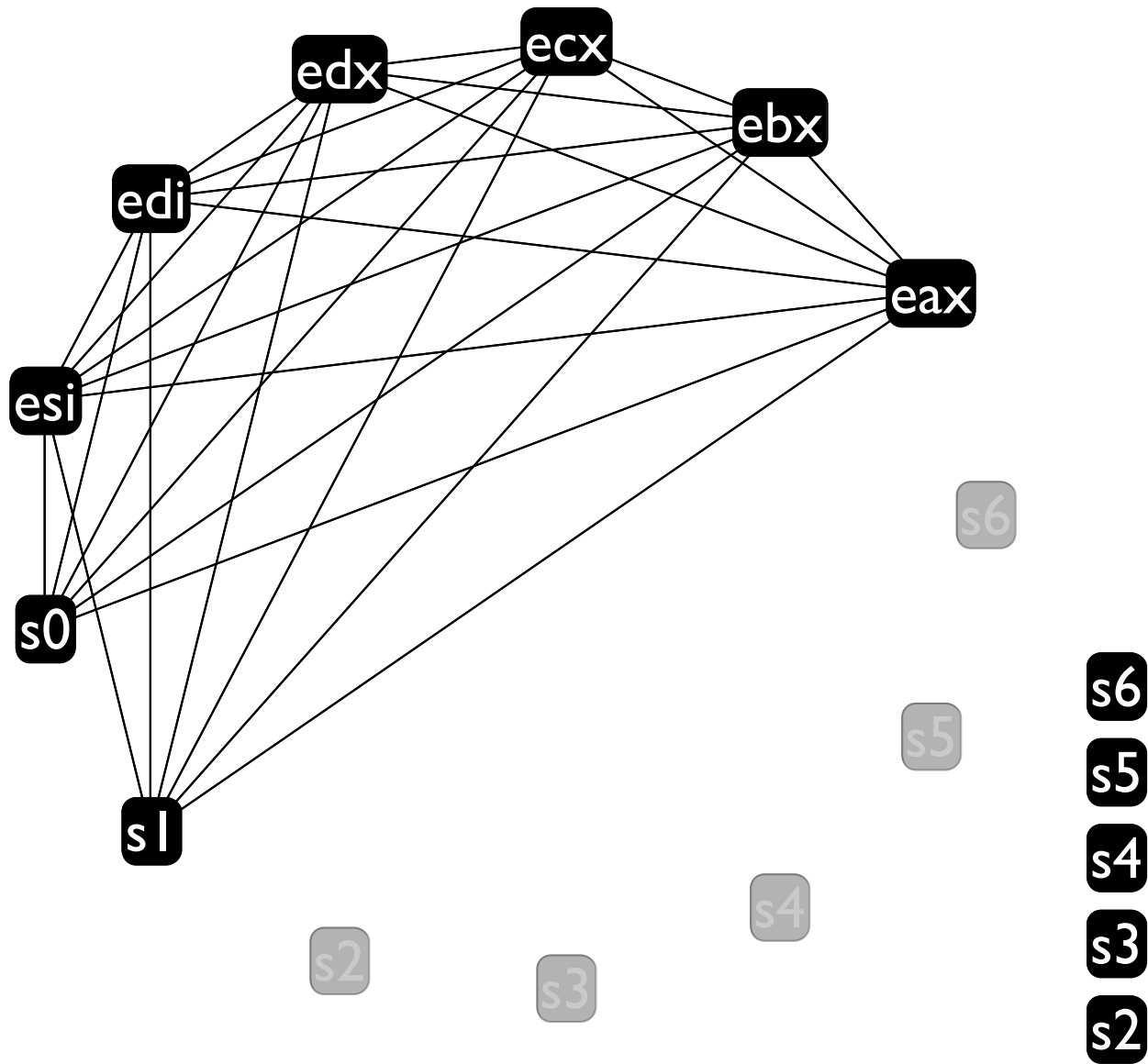




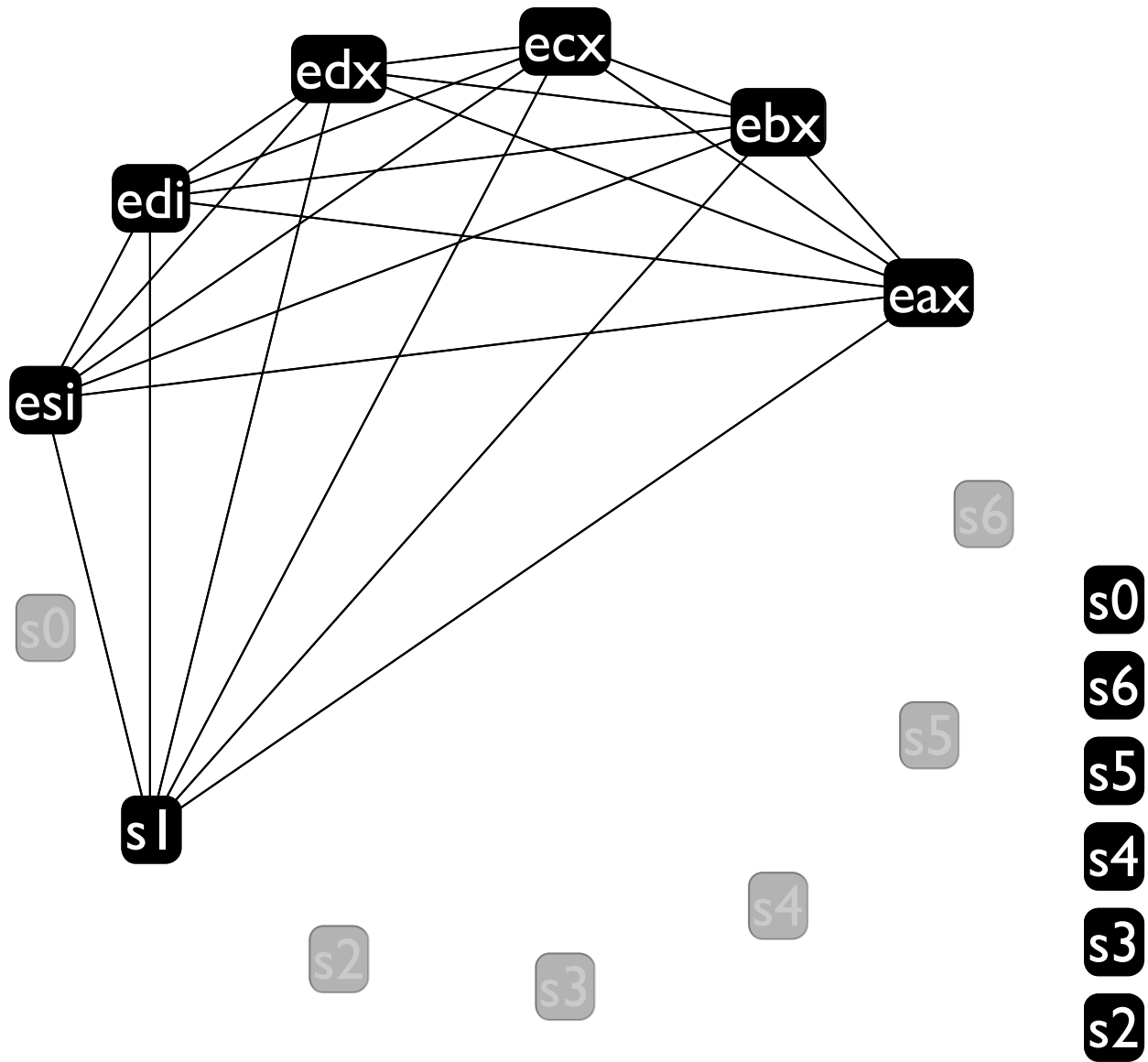


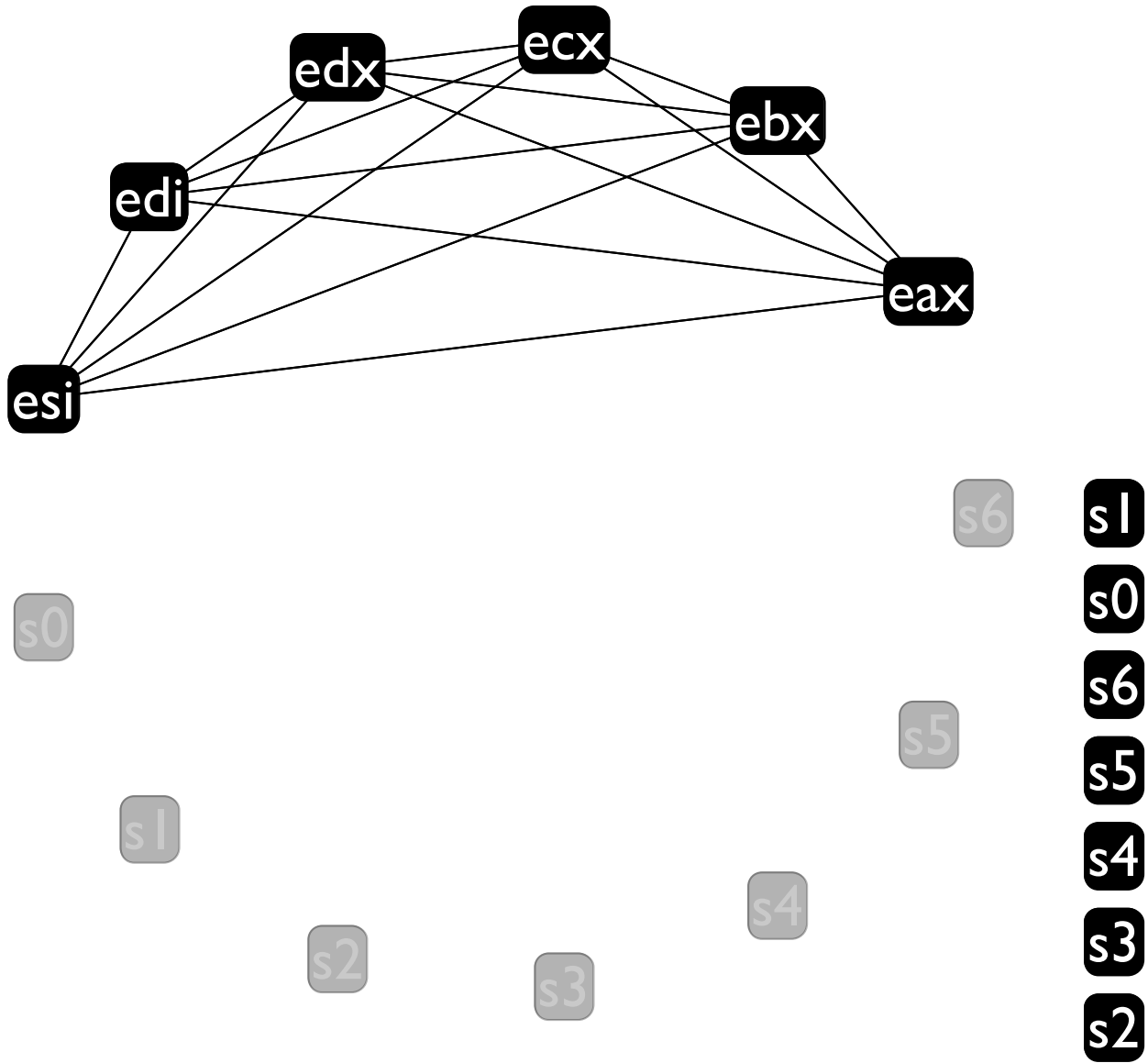


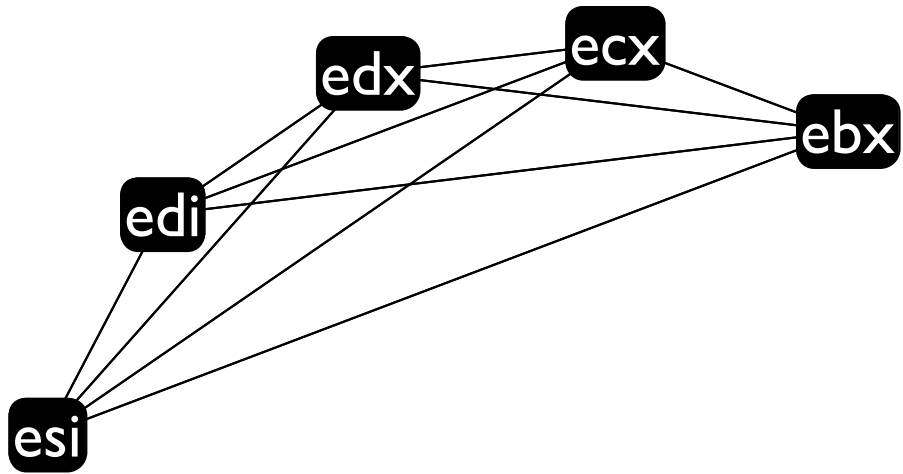












eax

**eax**

s6

**sl**

s0

**s0**

s5

**s6**

sl

**s5**

s2

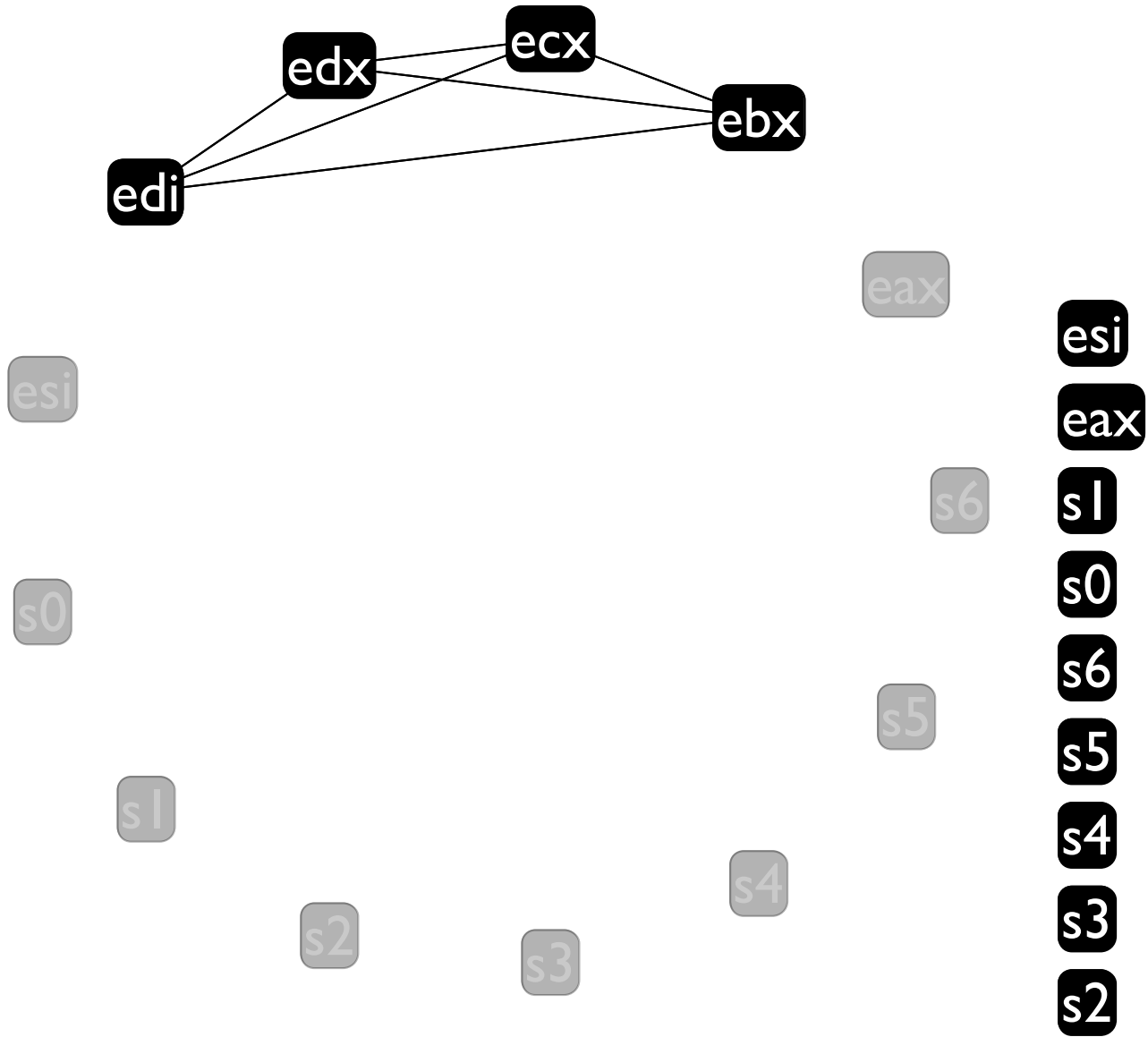
s4

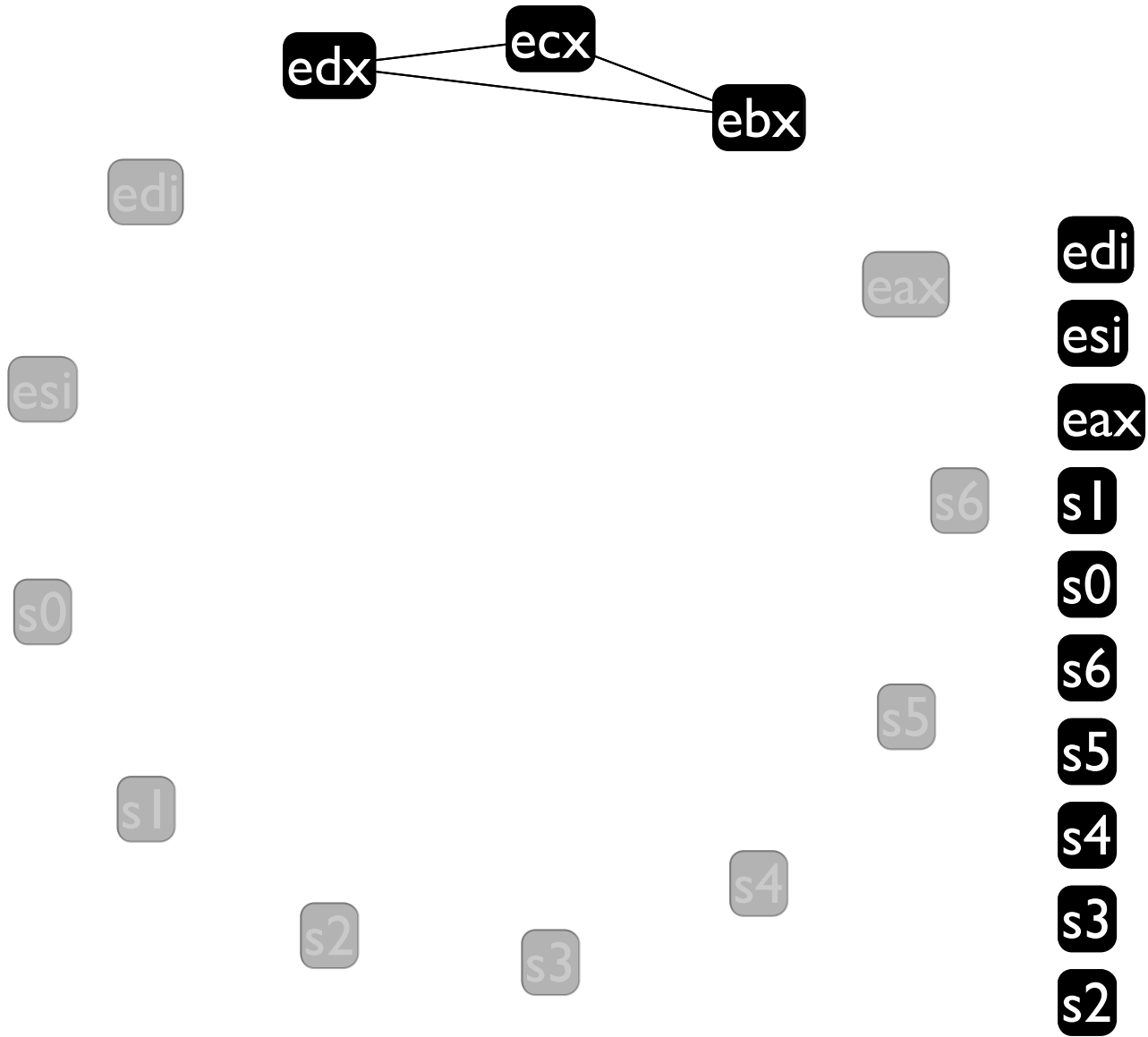
**s4**

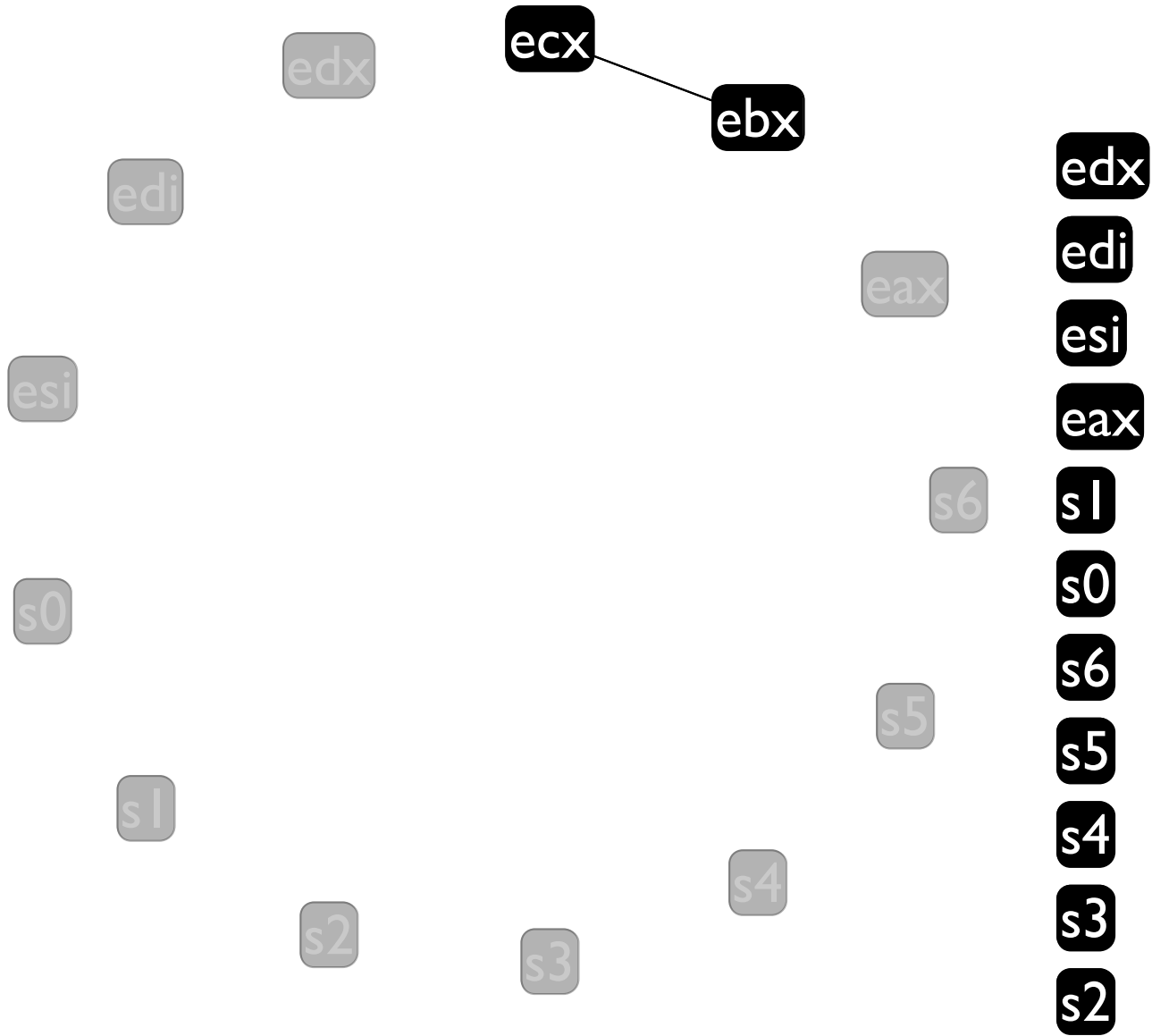
s3

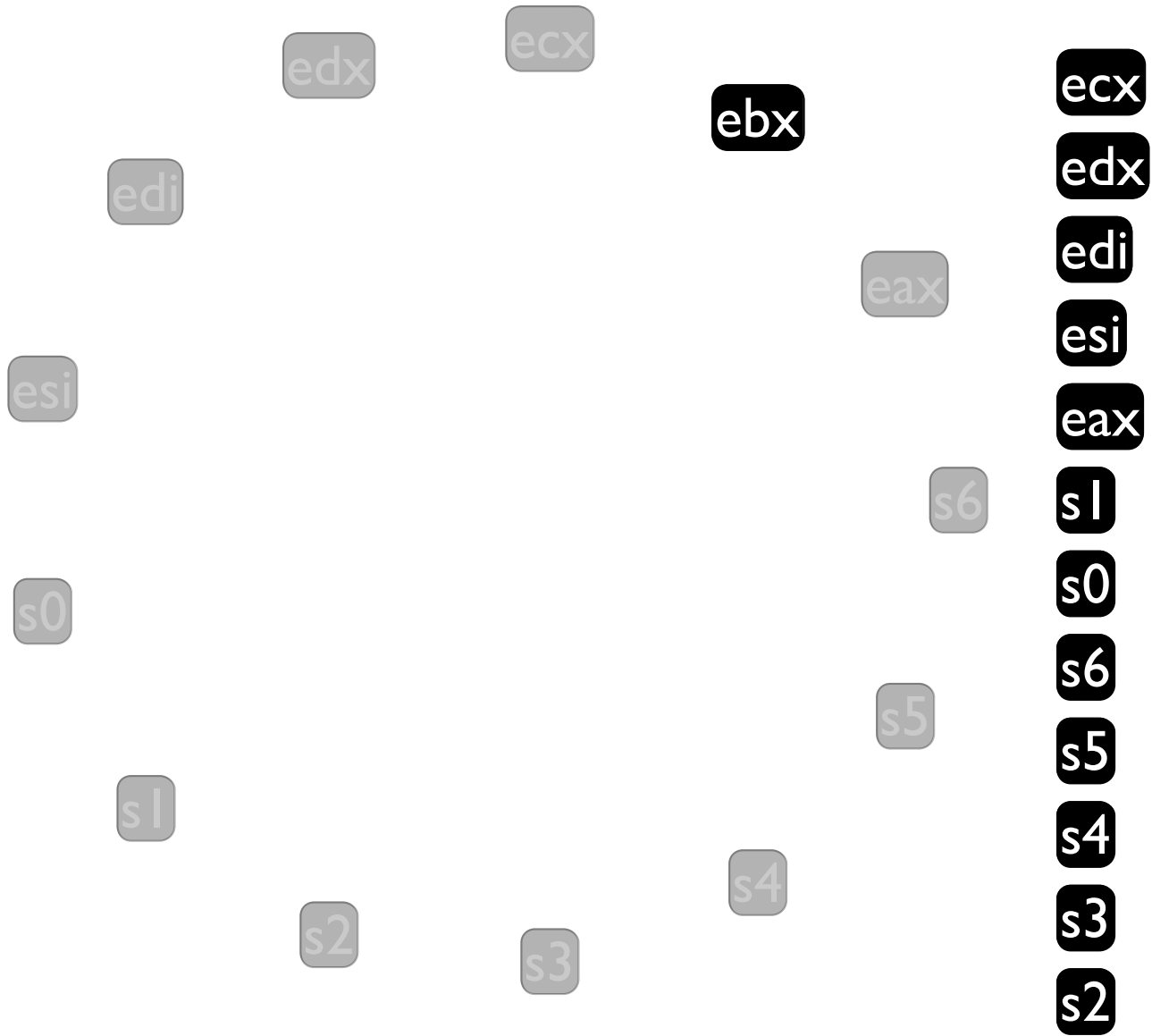
**s3**

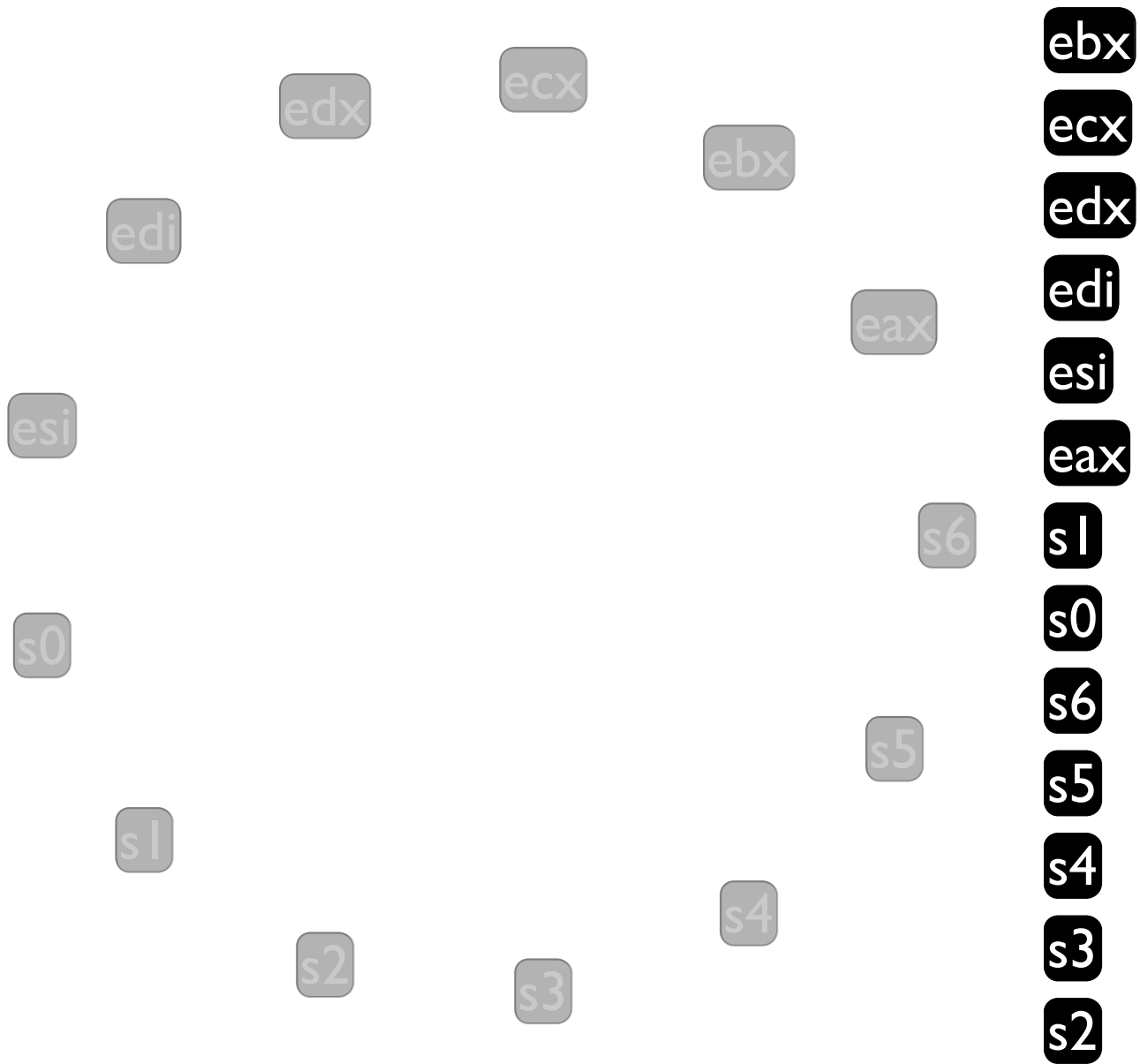
**s2**



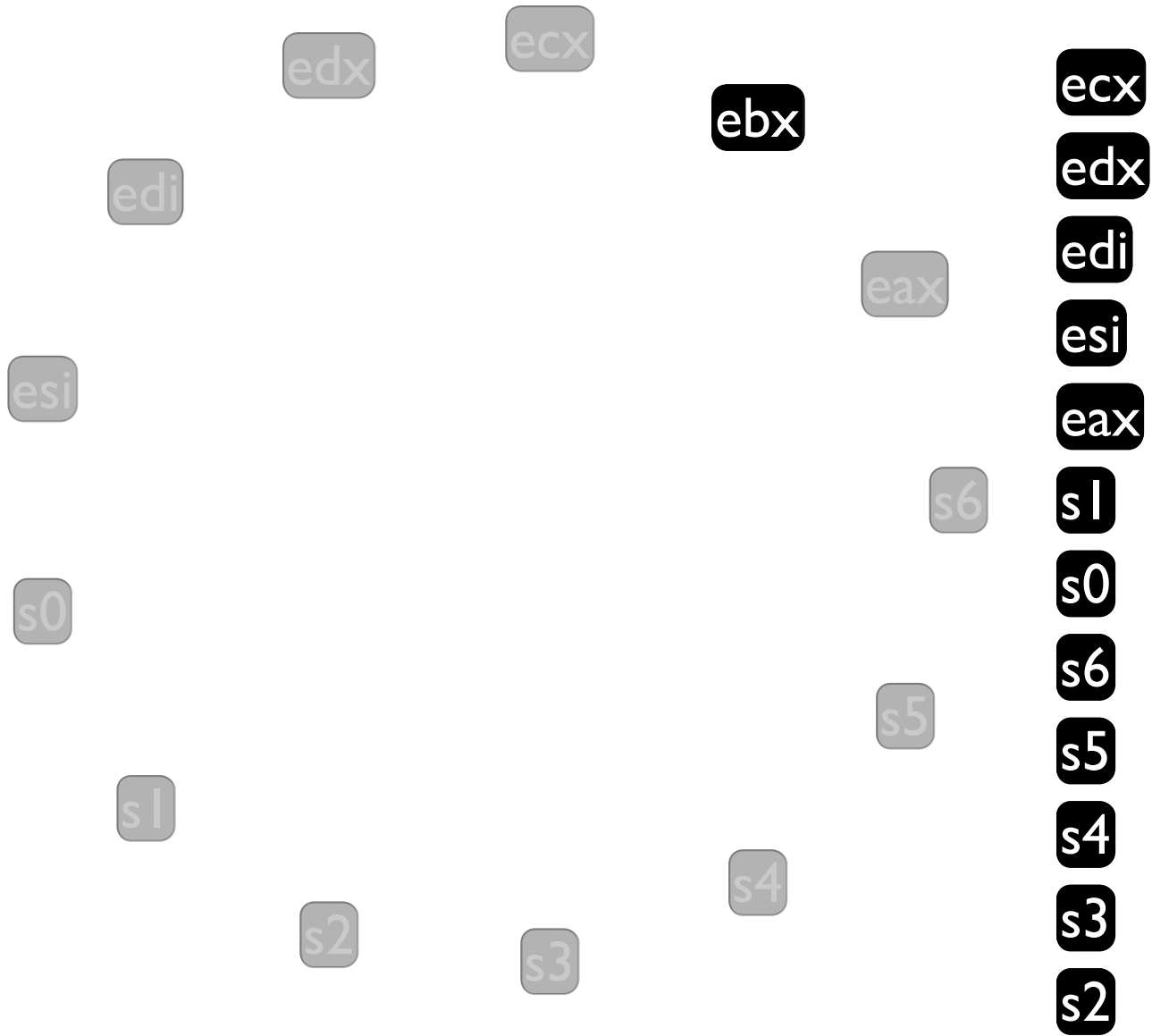


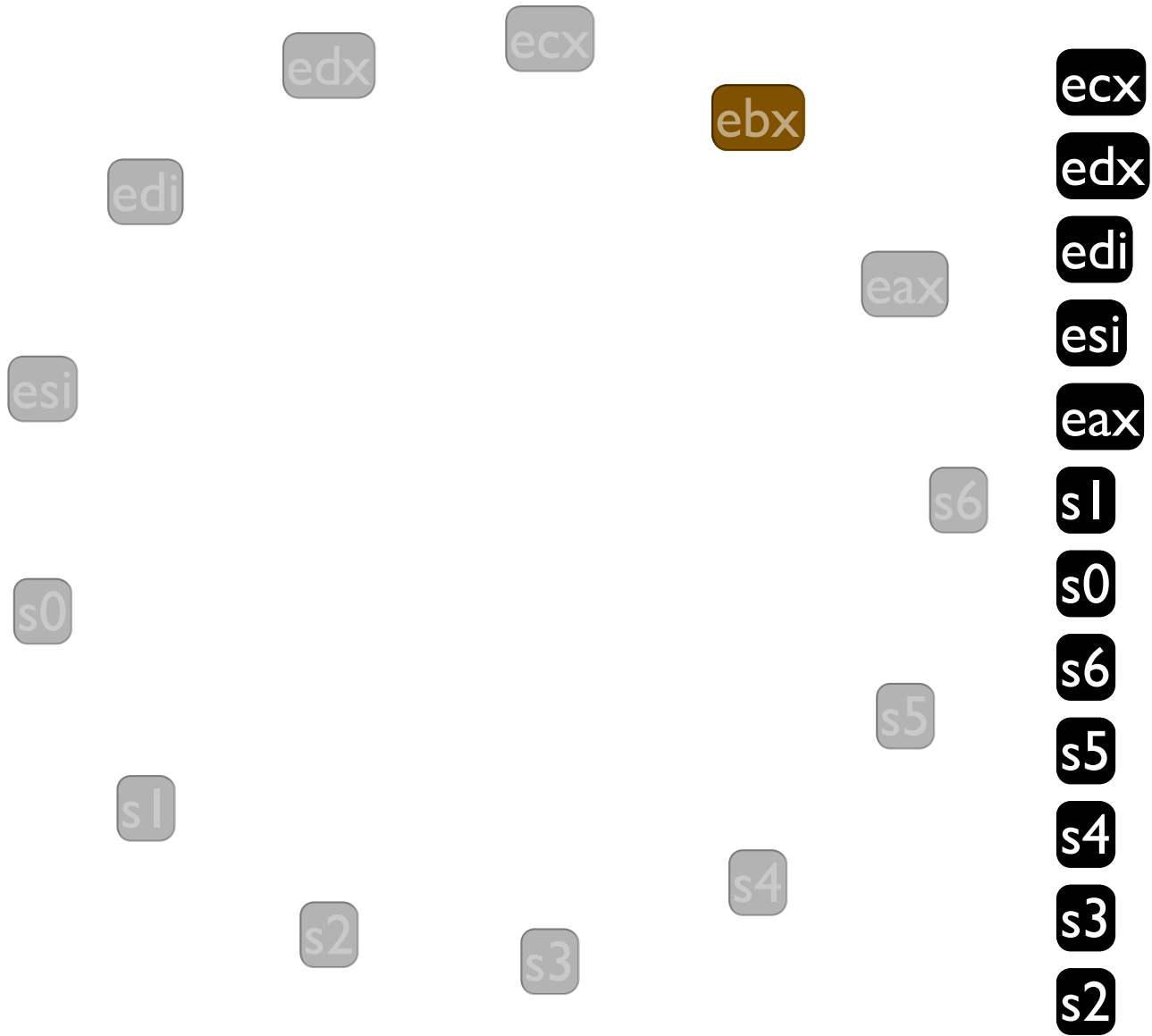


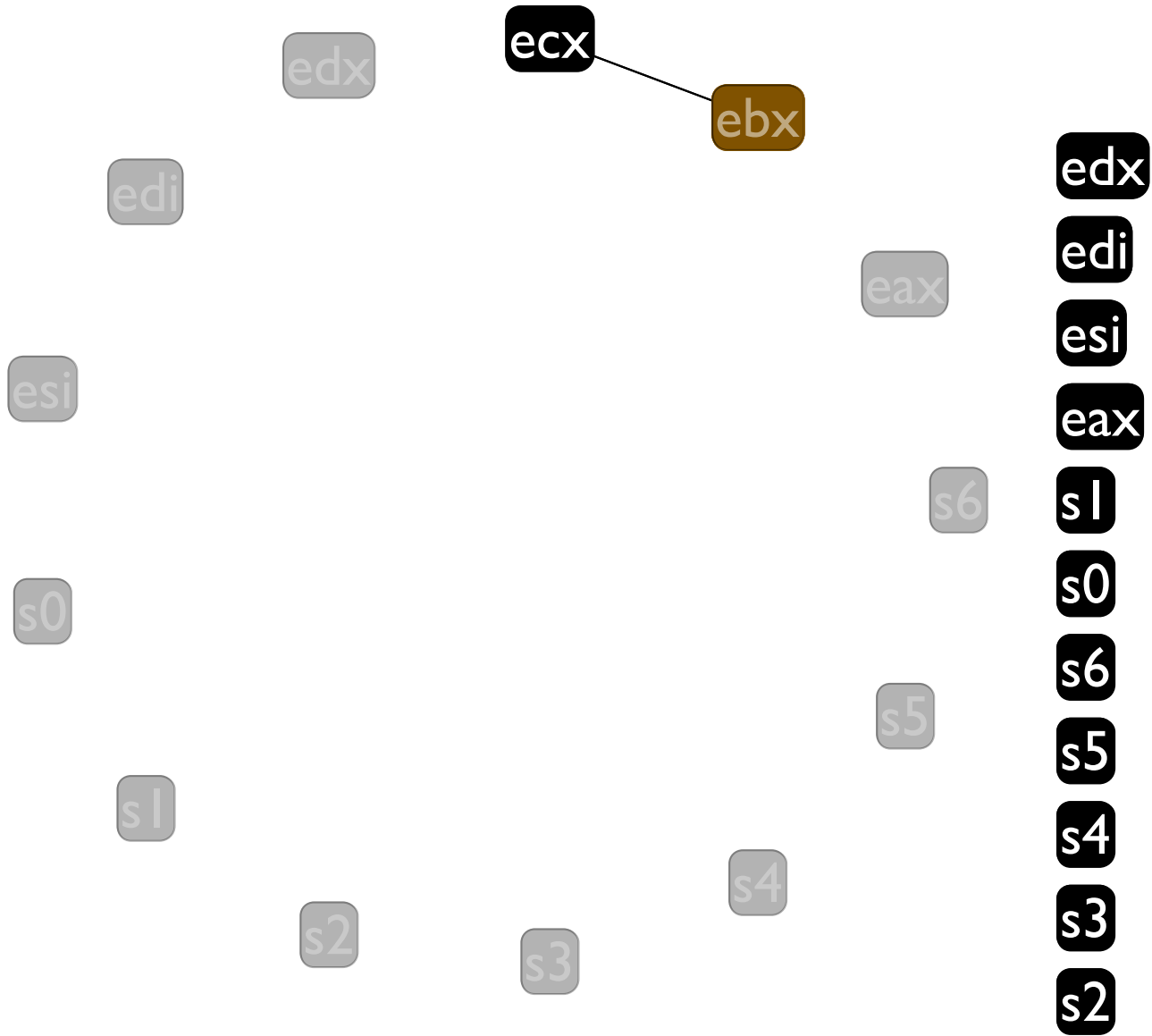


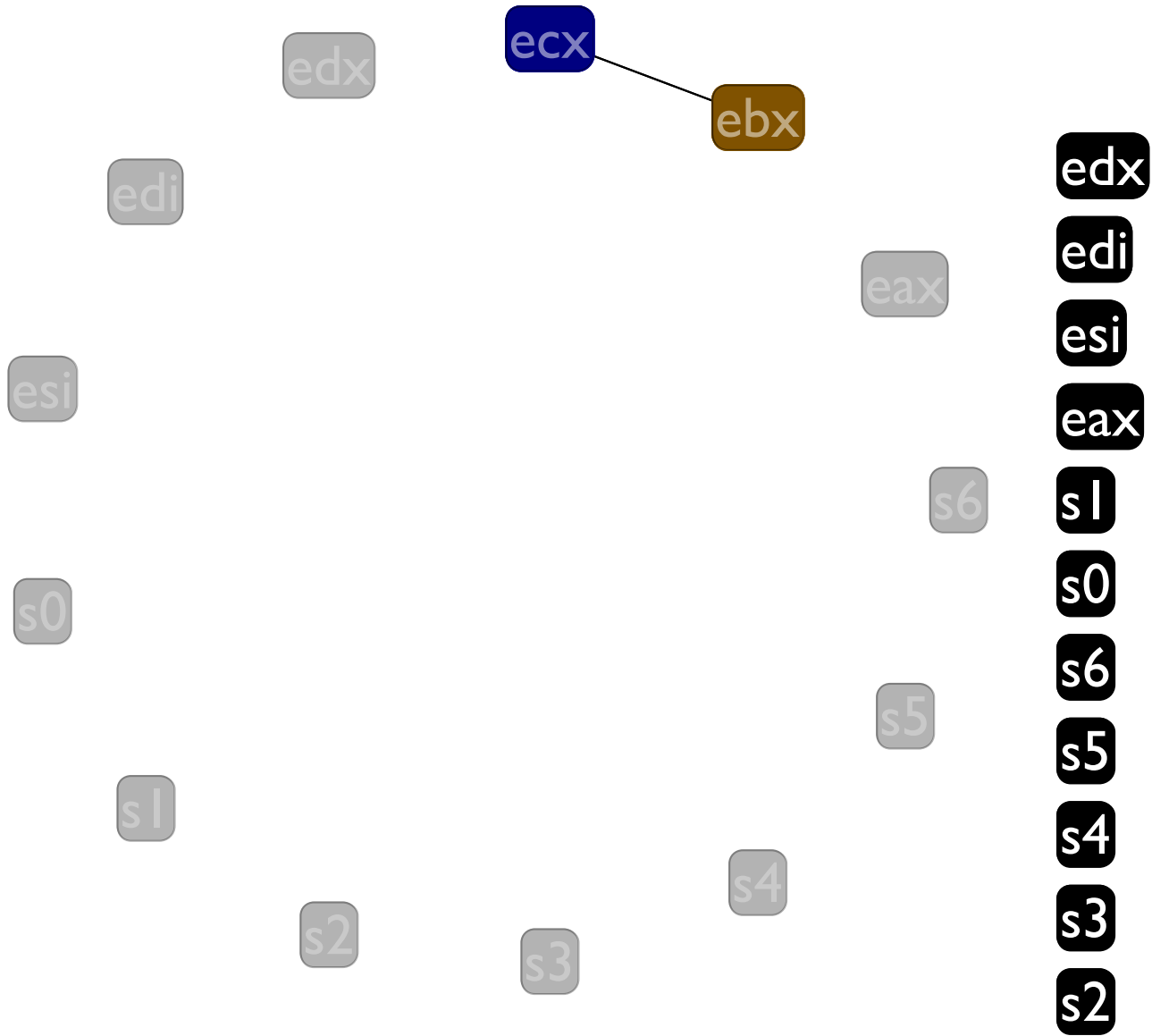


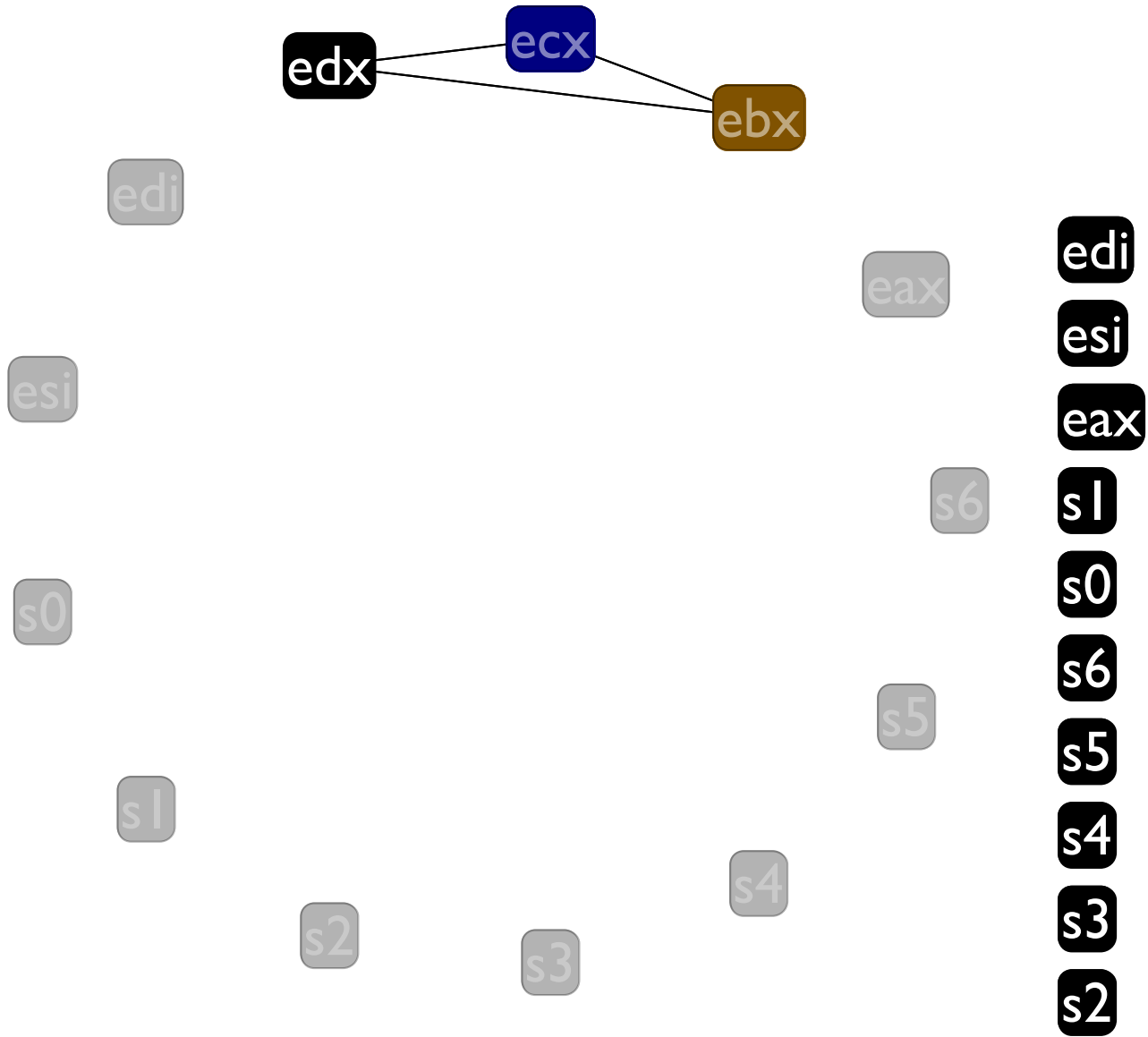


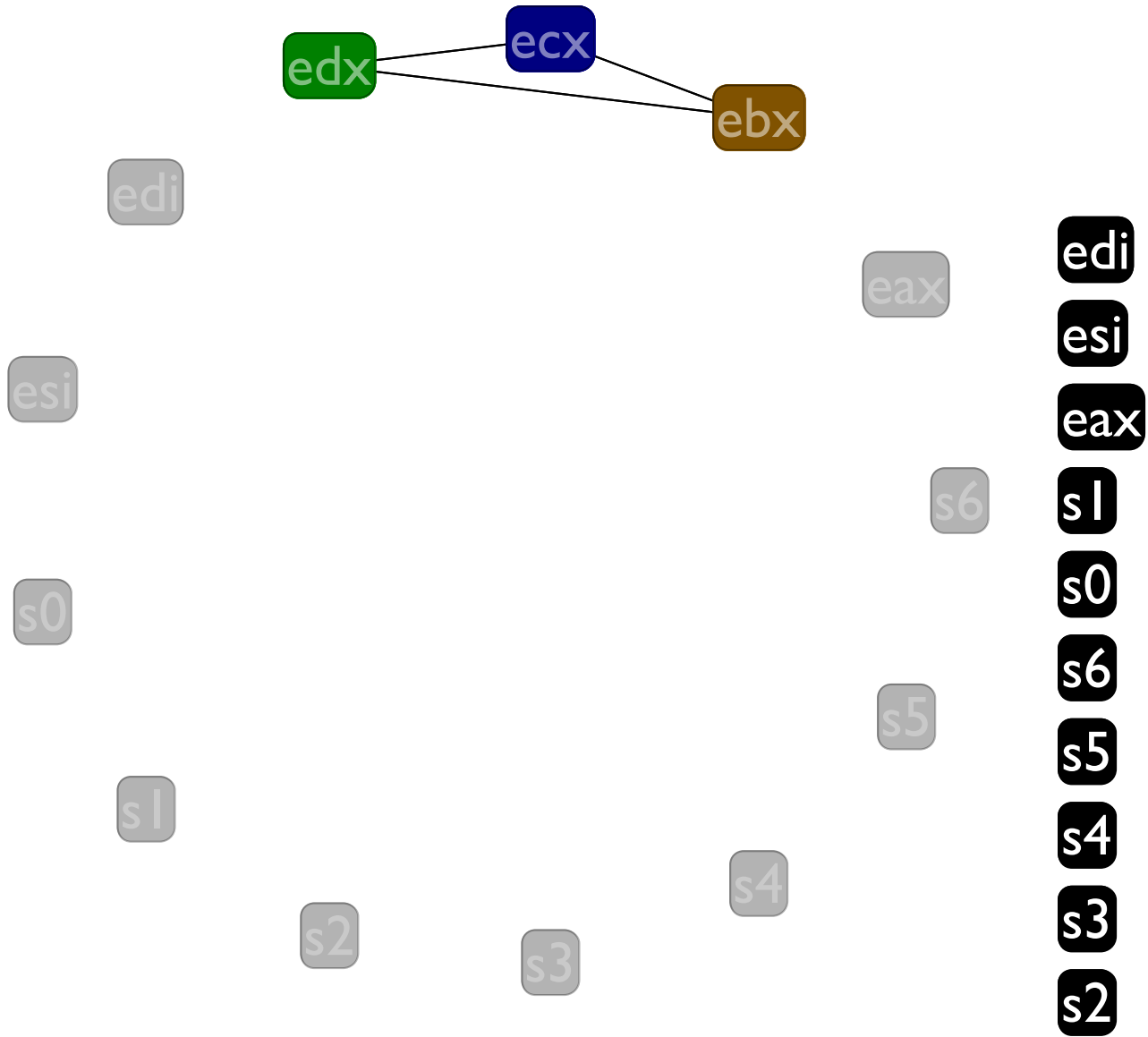


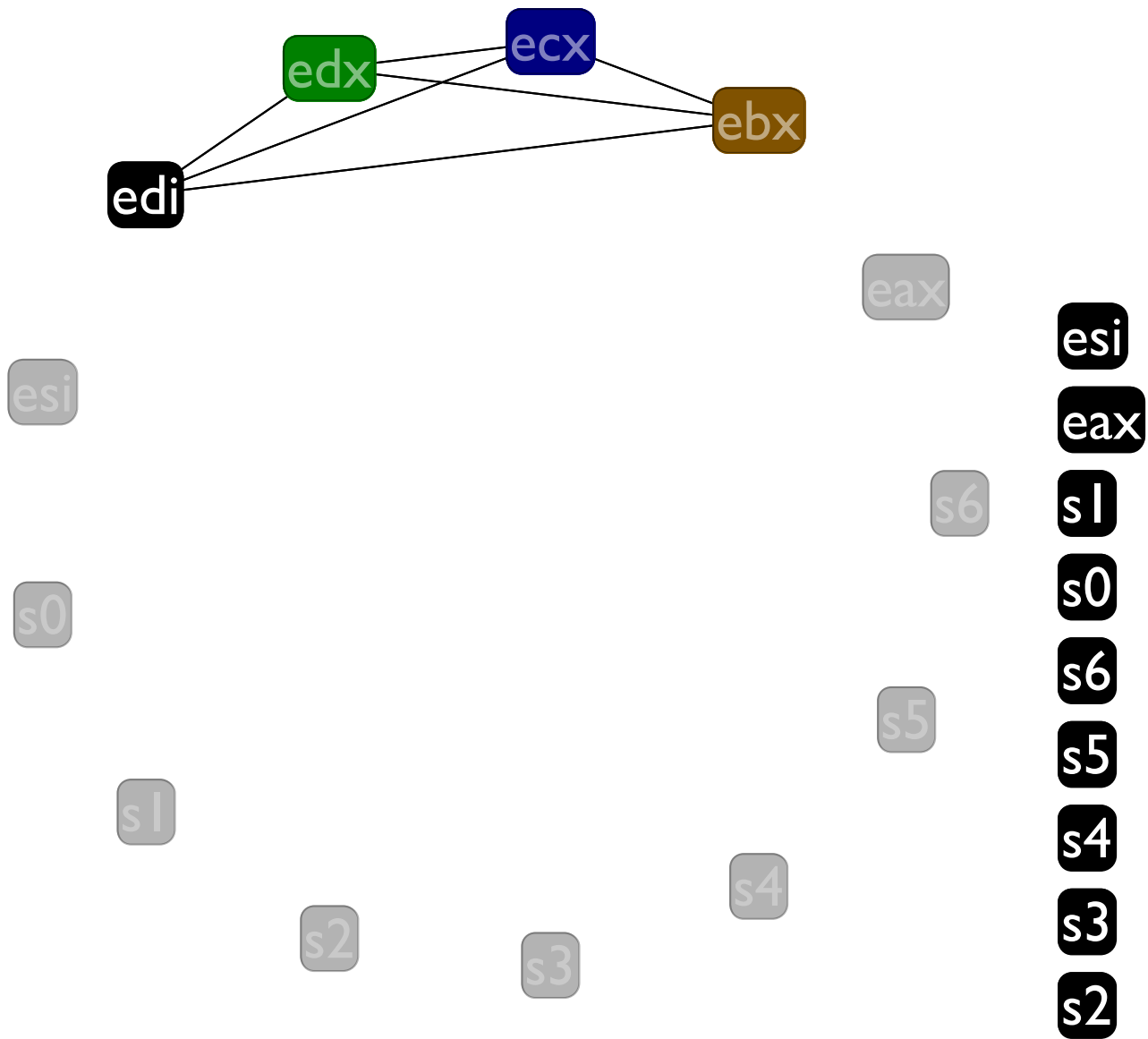


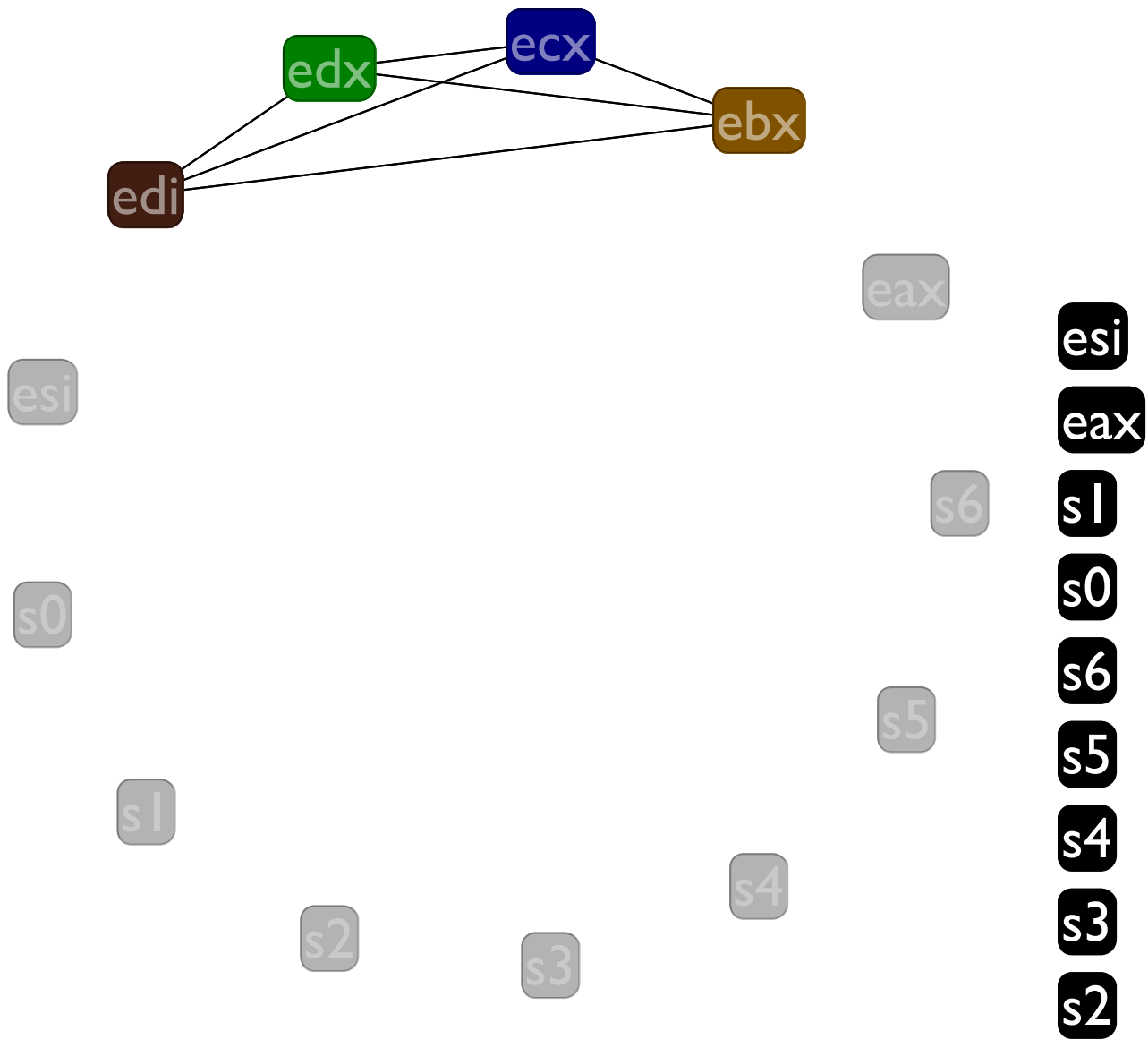




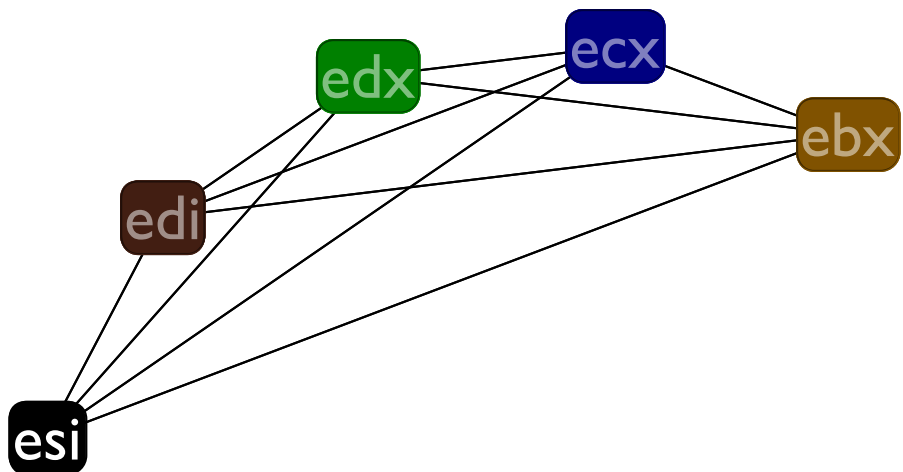












eax

s6

eax

s1

s0

s6

s5

s4

s3

s2

s0

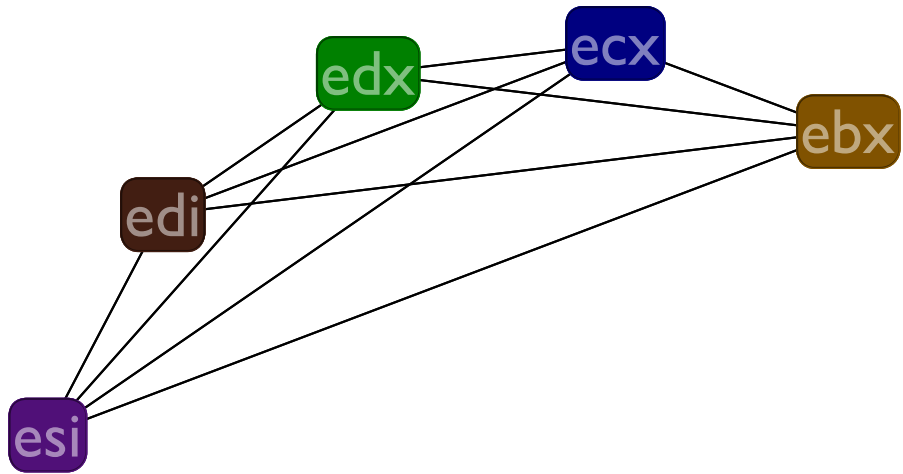
s1

s2

s3

s4

s5



eax

**eax**

s6

**s1**

s0

**s0**

s5

**s6**

s1

**s5**

s2

s4

**s4**

s3

**s3**

**s2**

