Register Allocation

Guido Wachsmuth

Overview
today's lecture

interference graphs

- construction during liveness analysis

Overview
today's lecture

interference graphs

- construction during liveness analysis

graph colouring

- assign registers to local variables and compiler temporaries
- store local variables and temporaries in memory

Overview
today's lecture

interference graphs

- construction during liveness analysis

graph colouring

- assign registers to local variables and compiler temporaries
- store local variables and temporaries in memory

coalescing

- handle move instructions

Overview
today's lecture

interference graphs

- construction during liveness analysis

graph colouring

- assign registers to local variables and compiler temporaries
- store local variables and temporaries in memory

coalescing

- handle move instructions

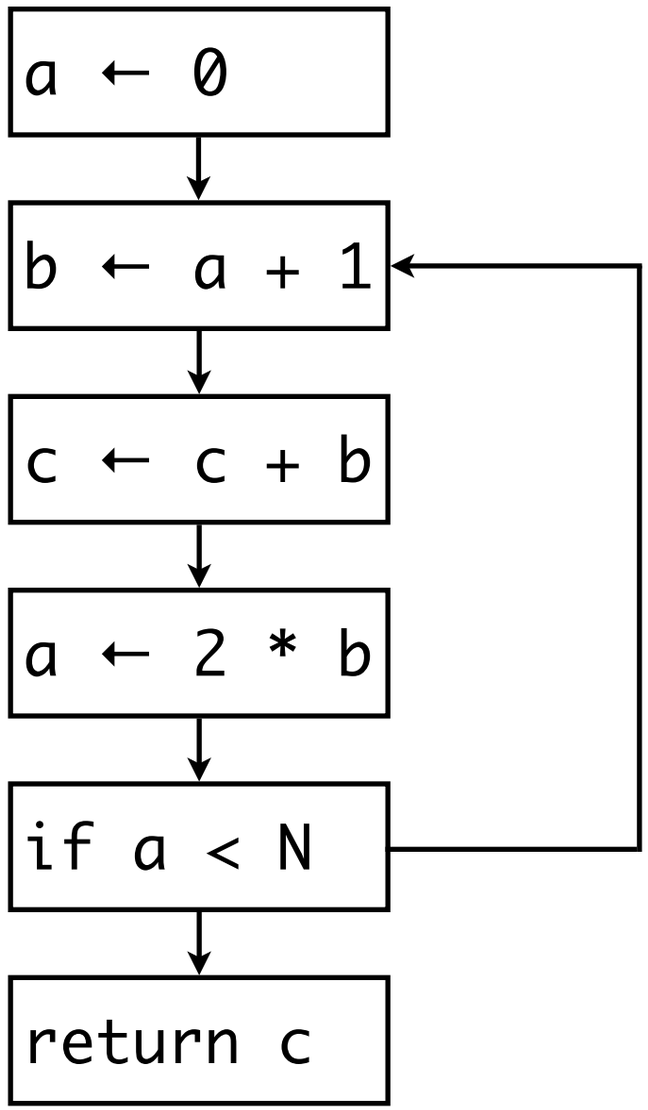
pre-coloured nodes

I

interference graphs

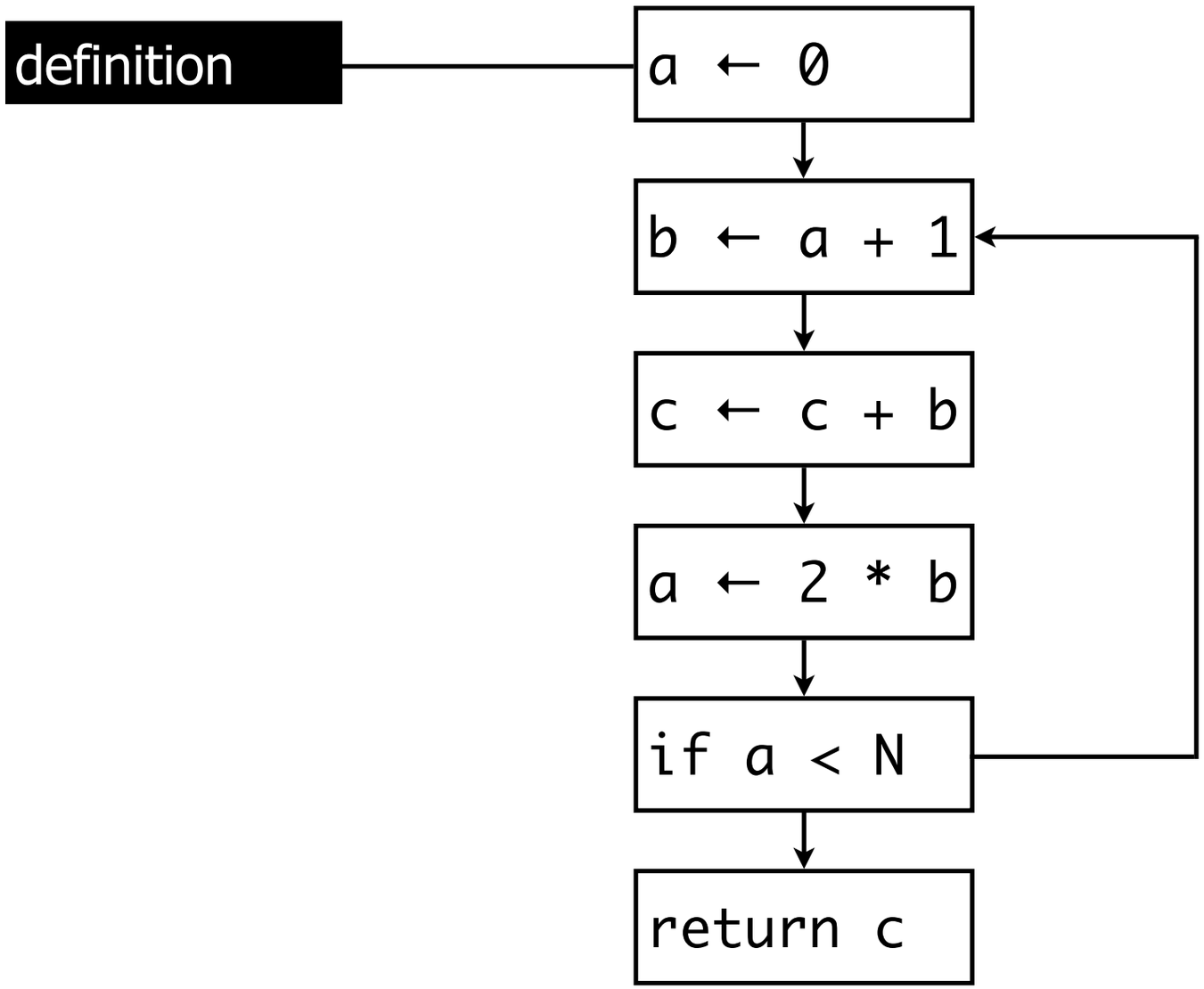
Recap: Liveness analysis

terminology



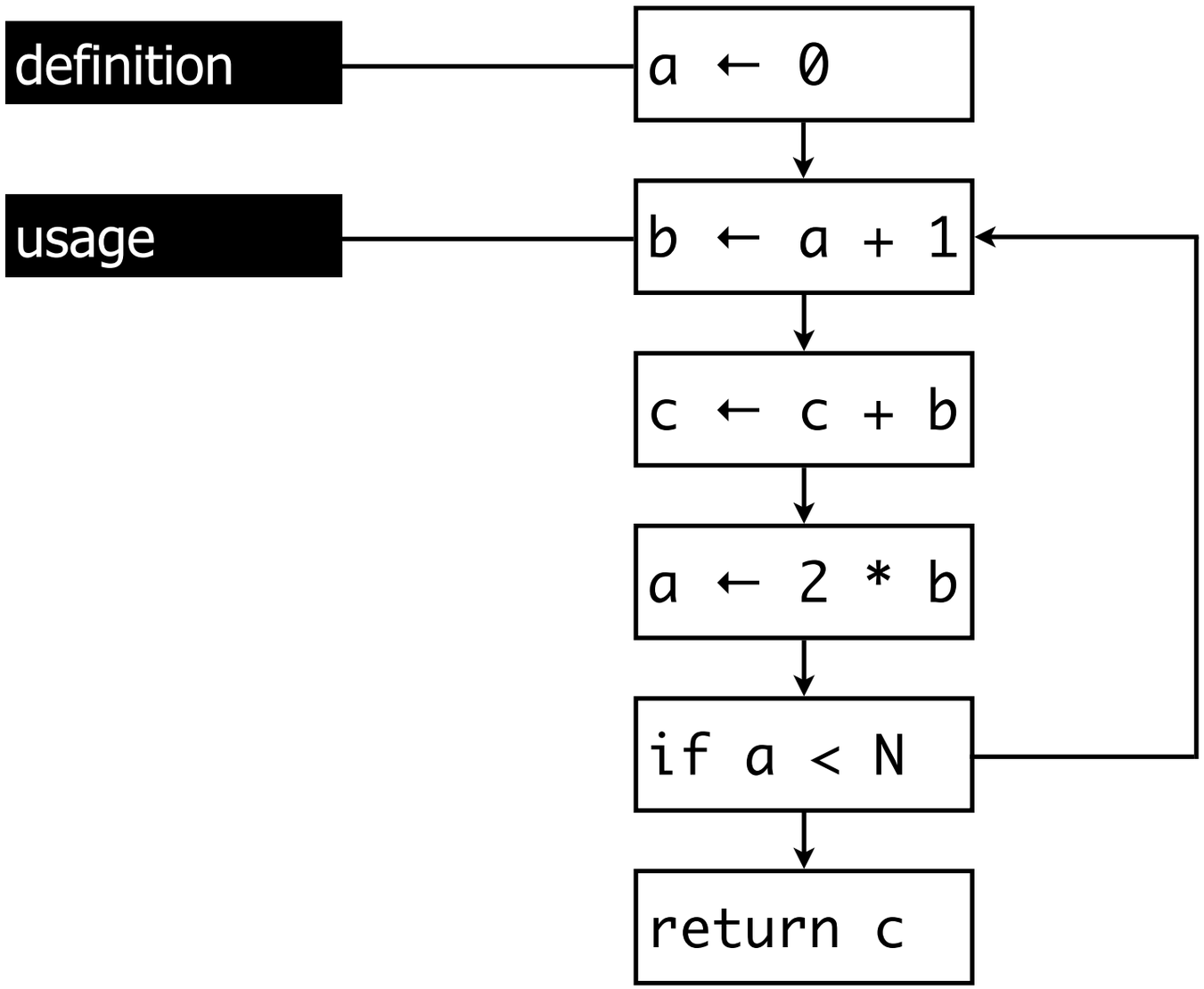
Recap: Liveness analysis

terminology



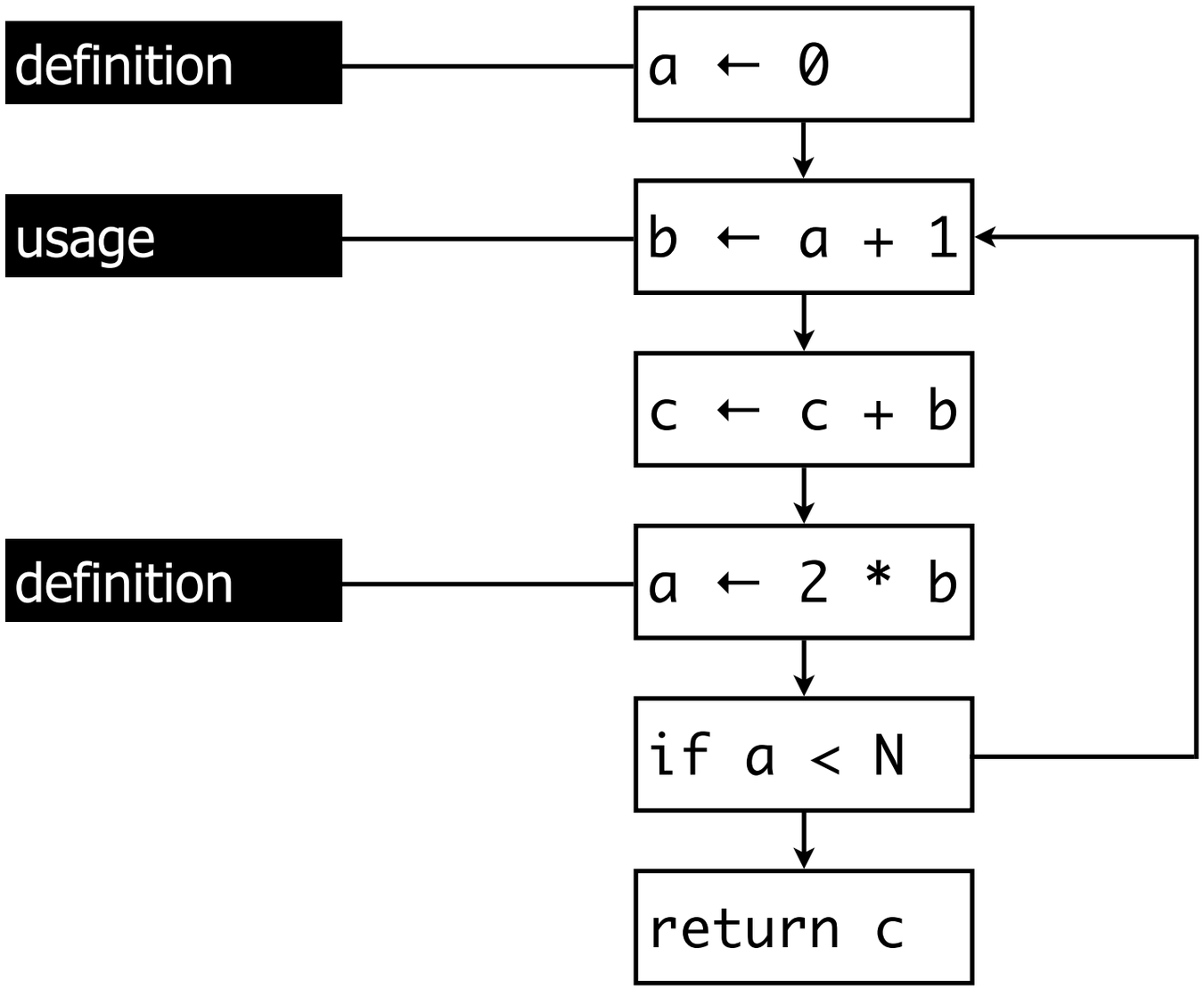
Recap: Liveness analysis

terminology



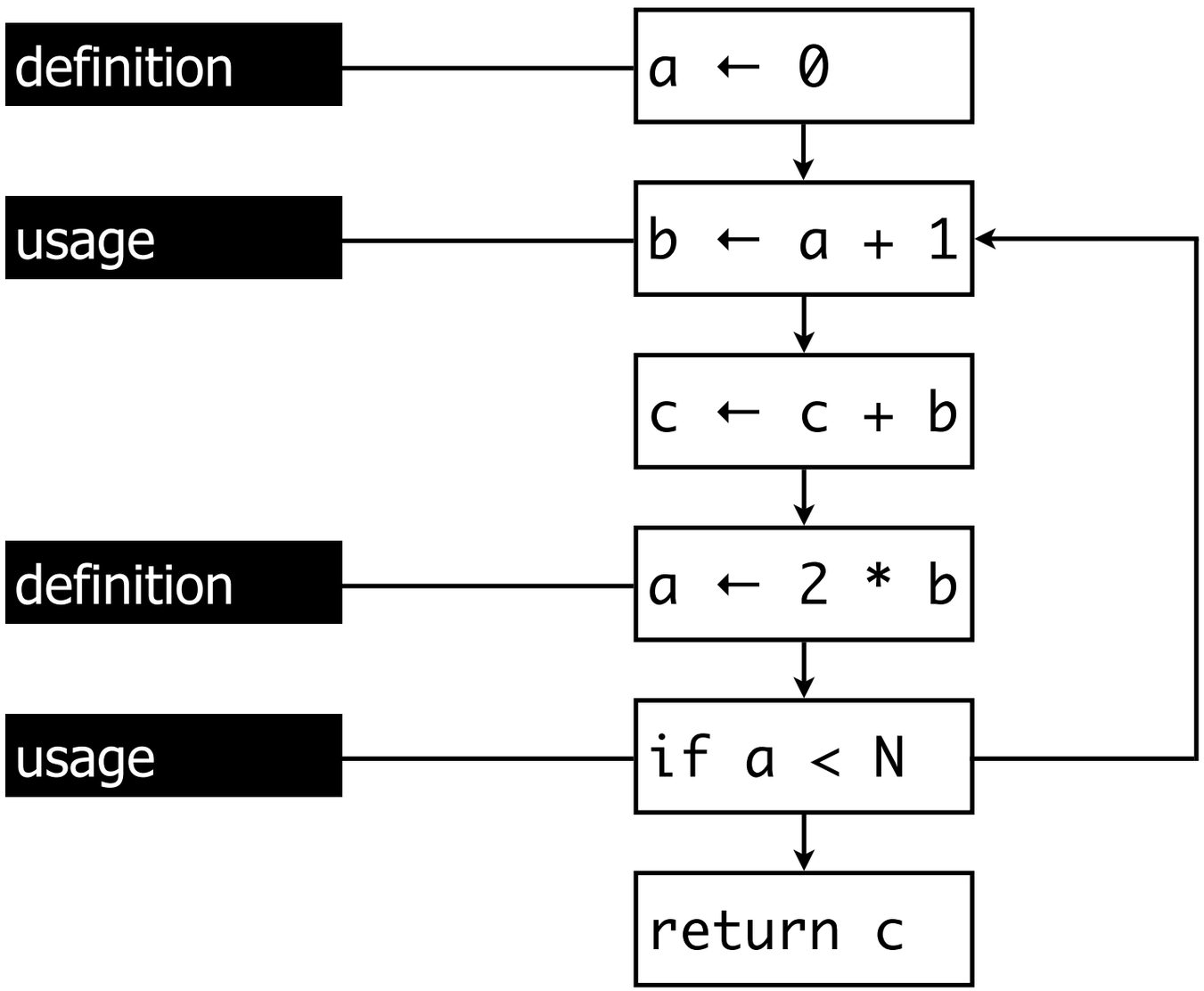
Recap: Liveness analysis

terminology



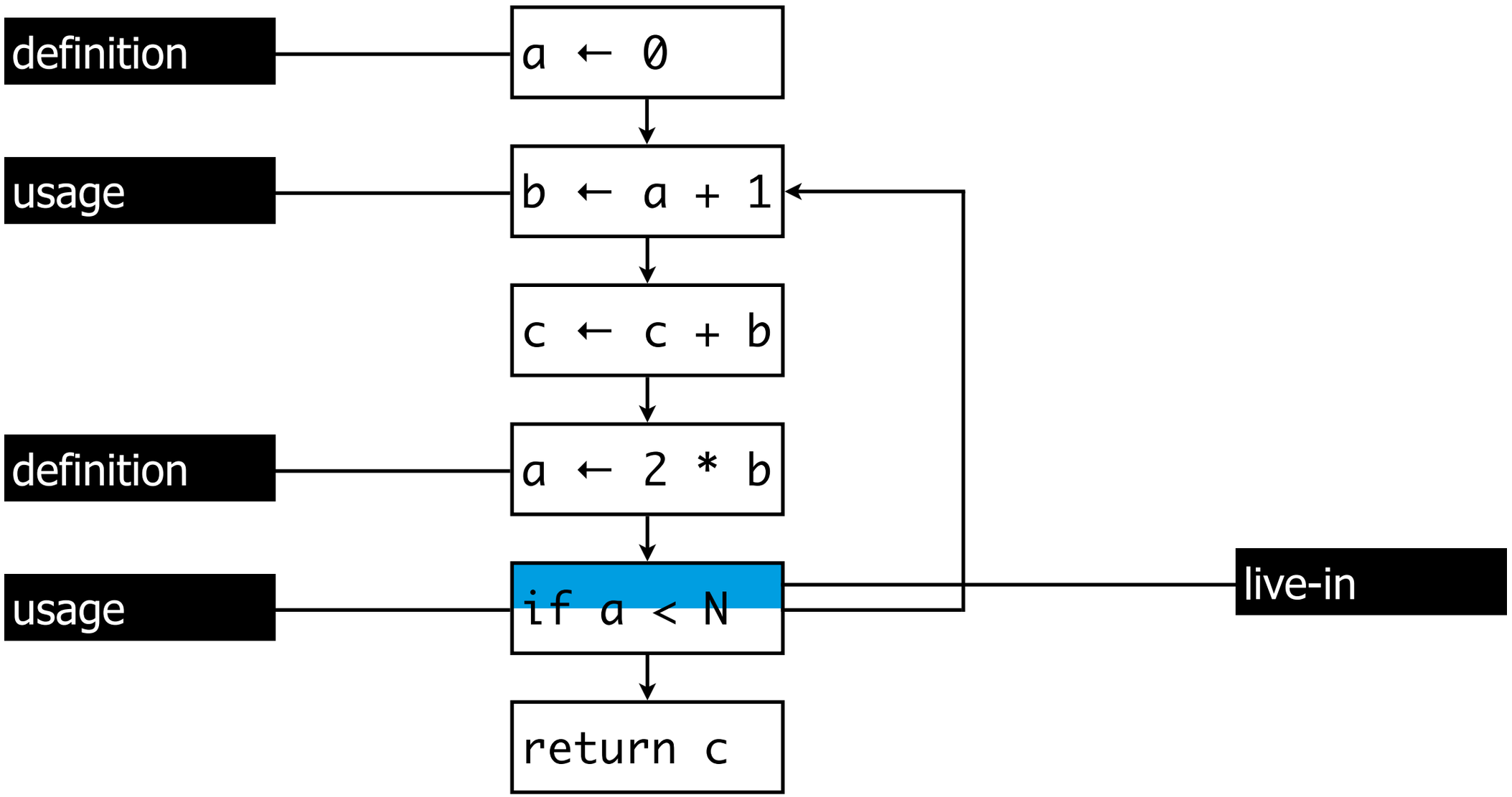
Recap: Liveness analysis

terminology



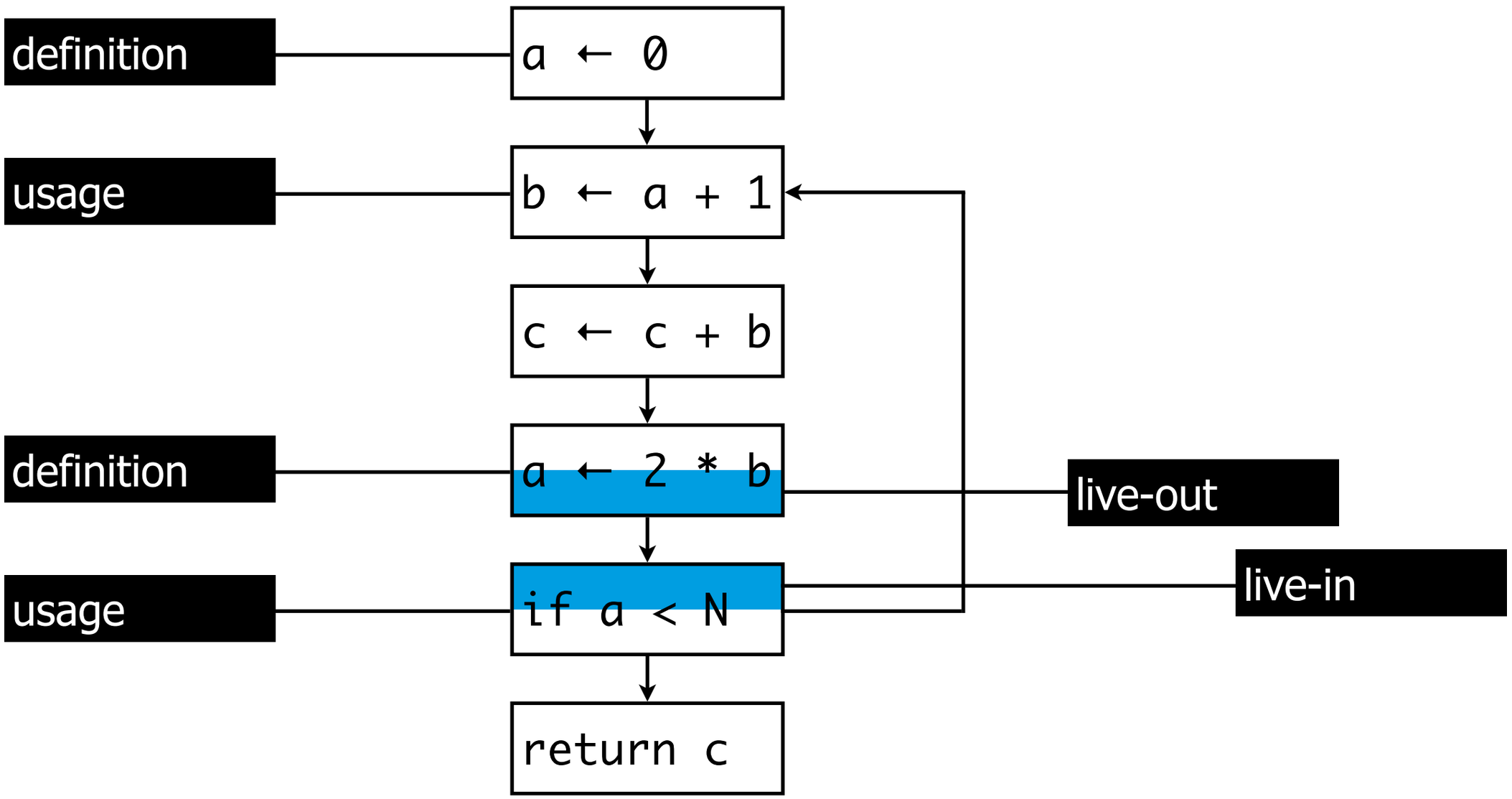
Recap: Liveness analysis

terminology



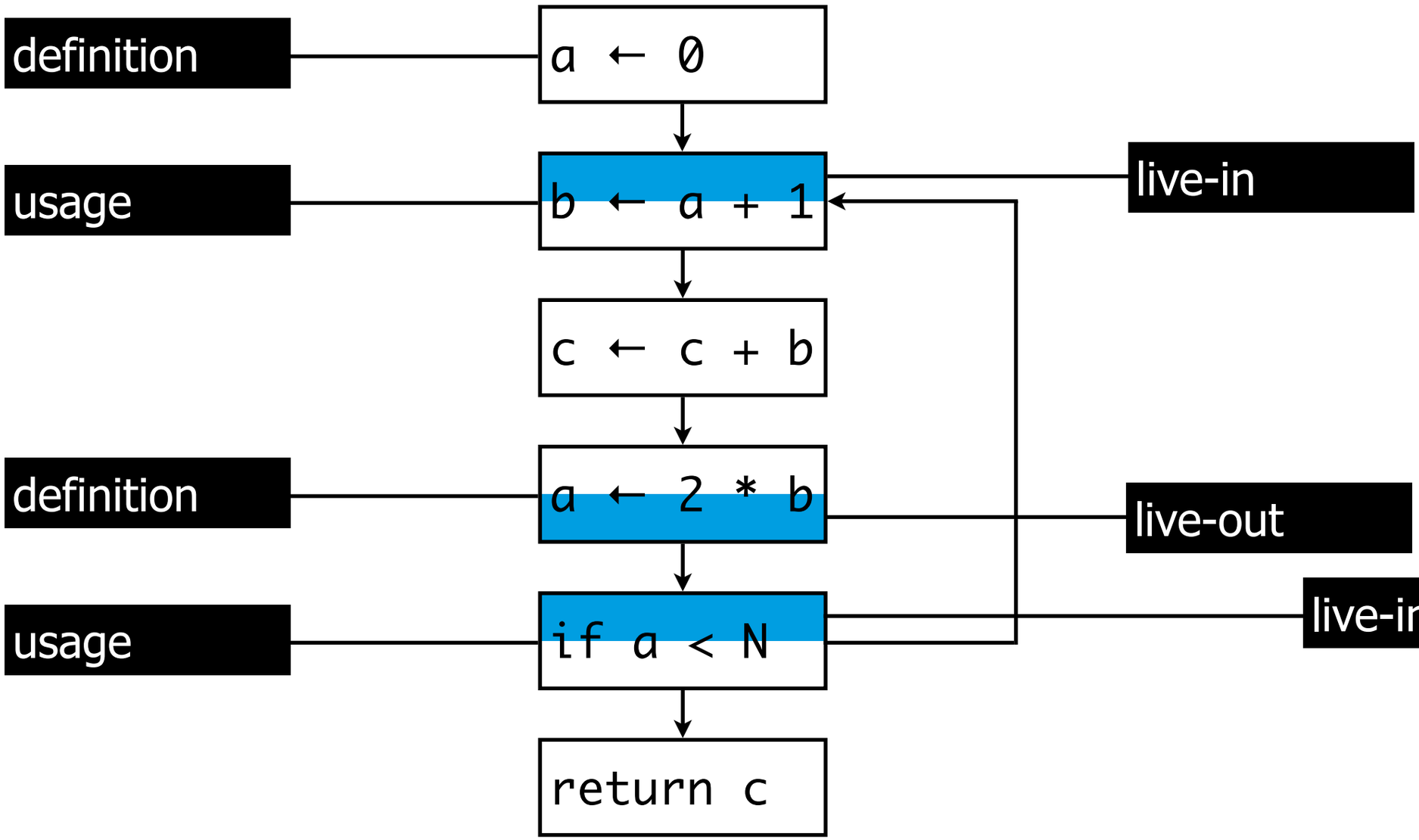
Recap: Liveness analysis

terminology



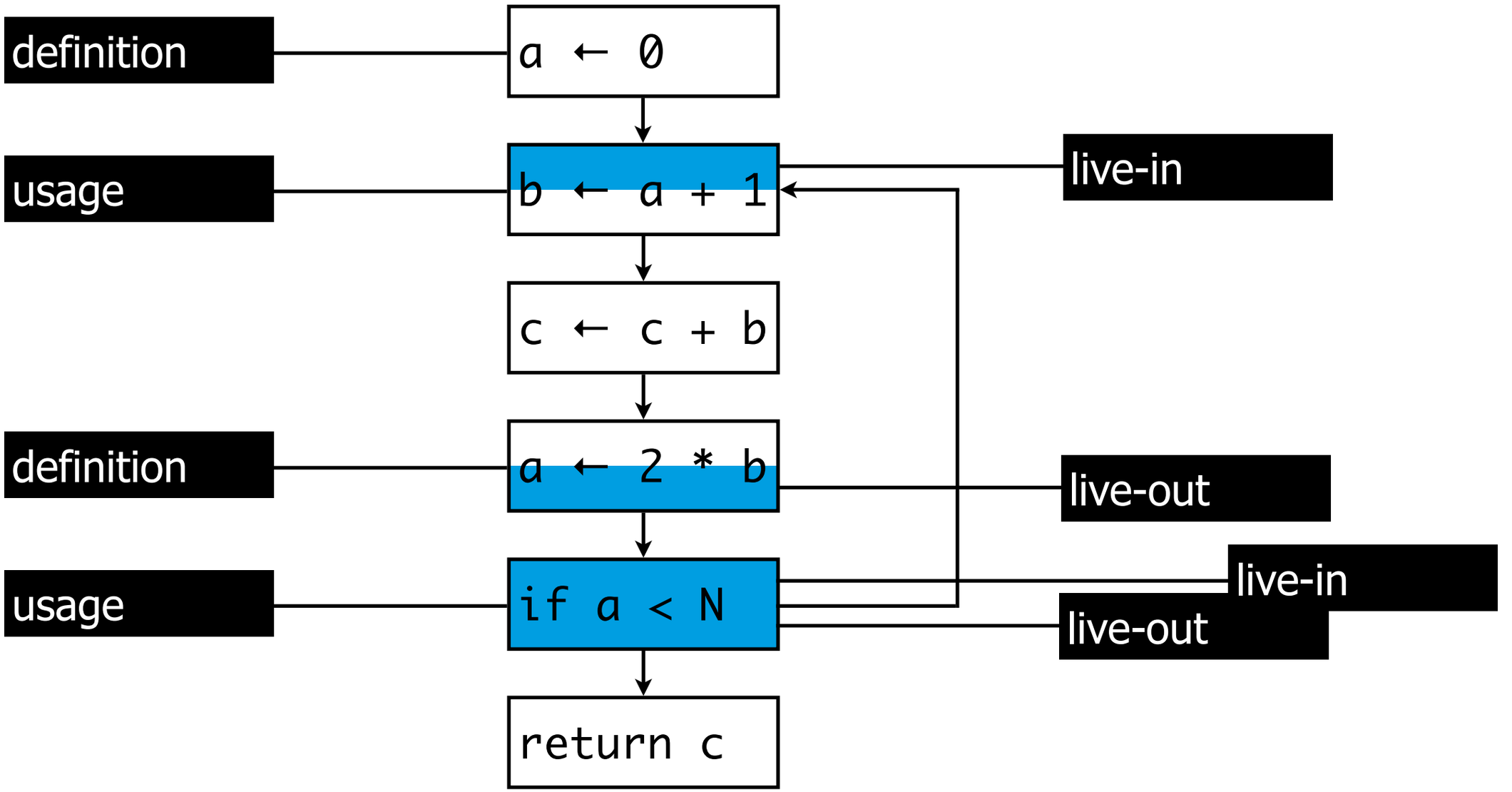
Recap: Liveness analysis

terminology



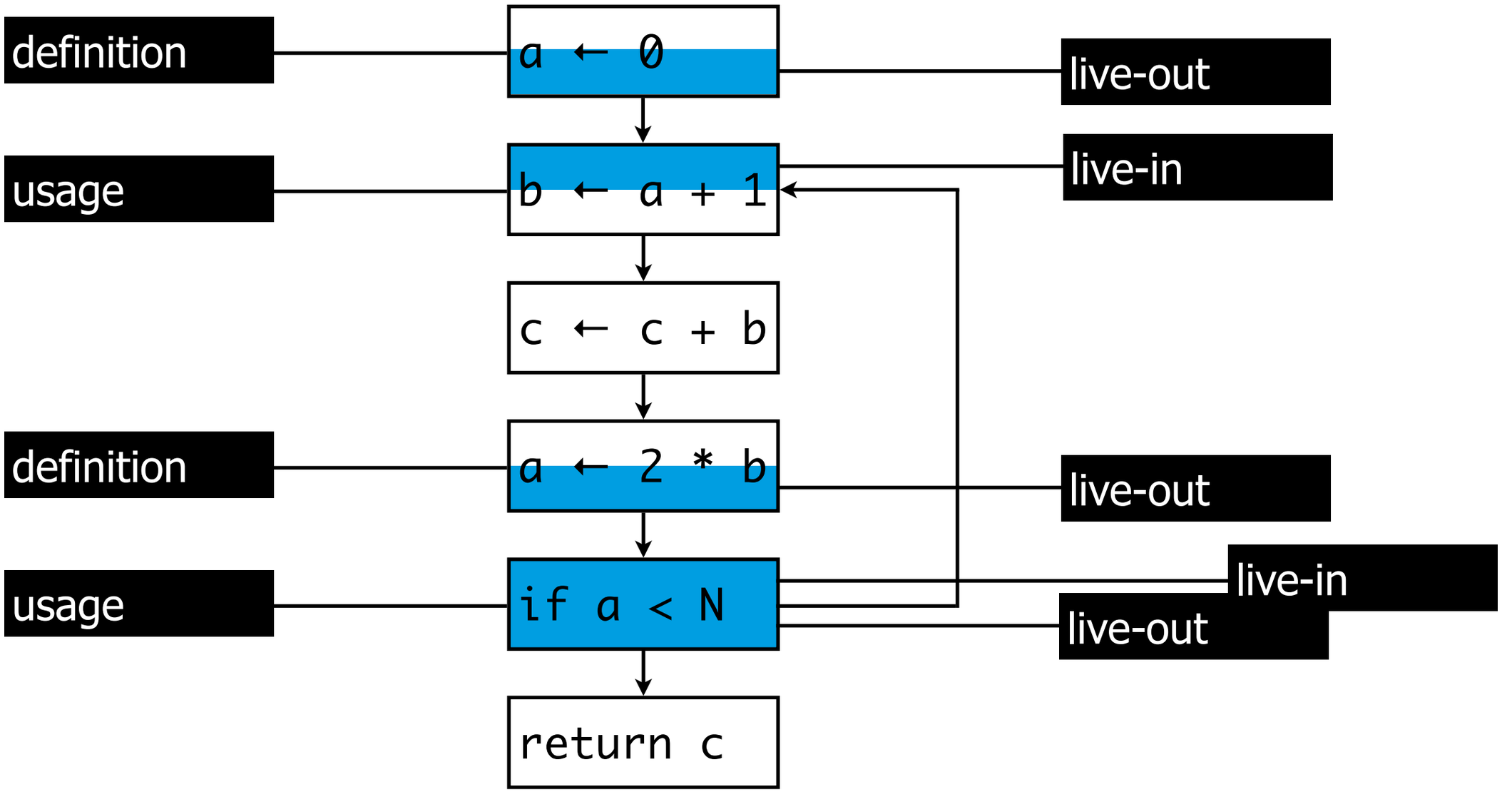
Recap: Liveness analysis

terminology



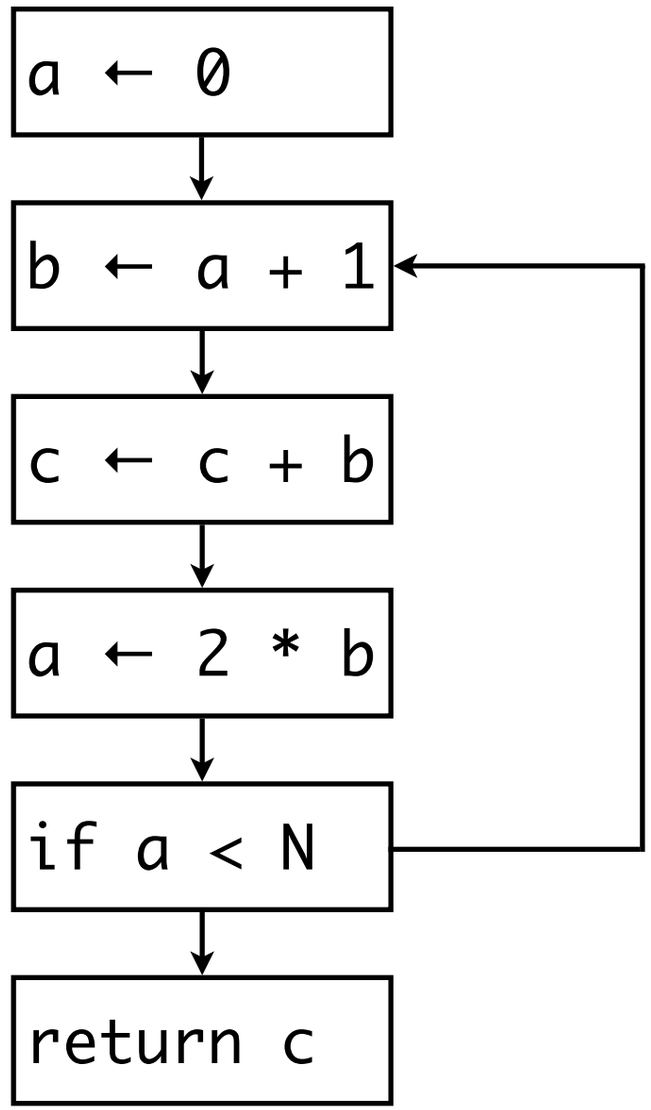
Recap: Liveness analysis

terminology



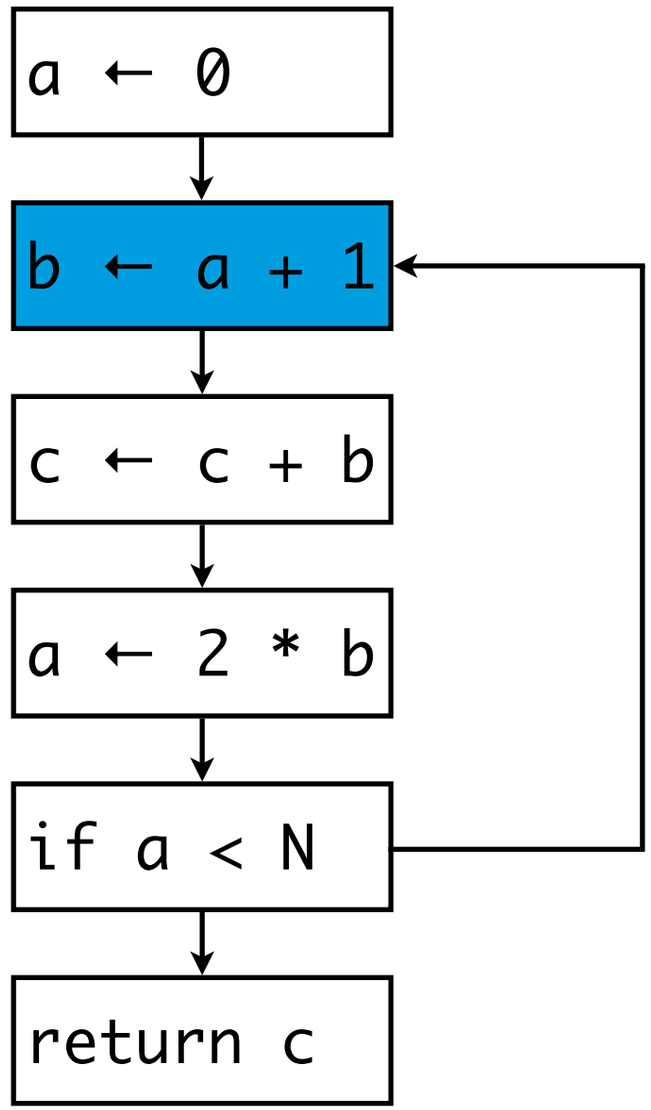
Recap: Liveness analysis

example



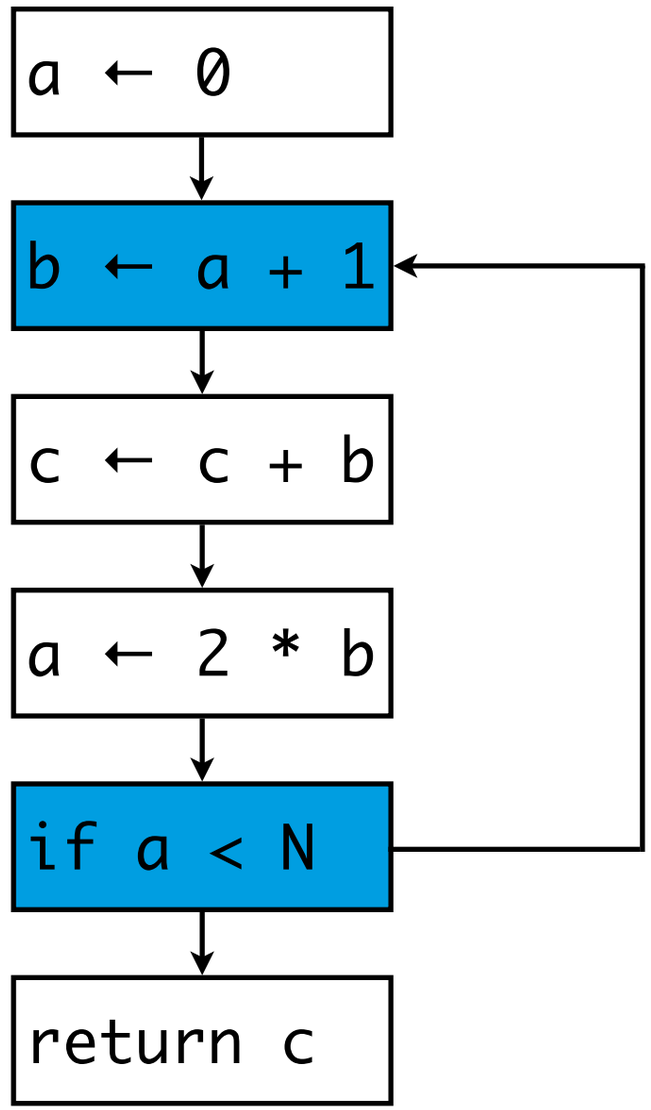
Recap: Liveness analysis

example



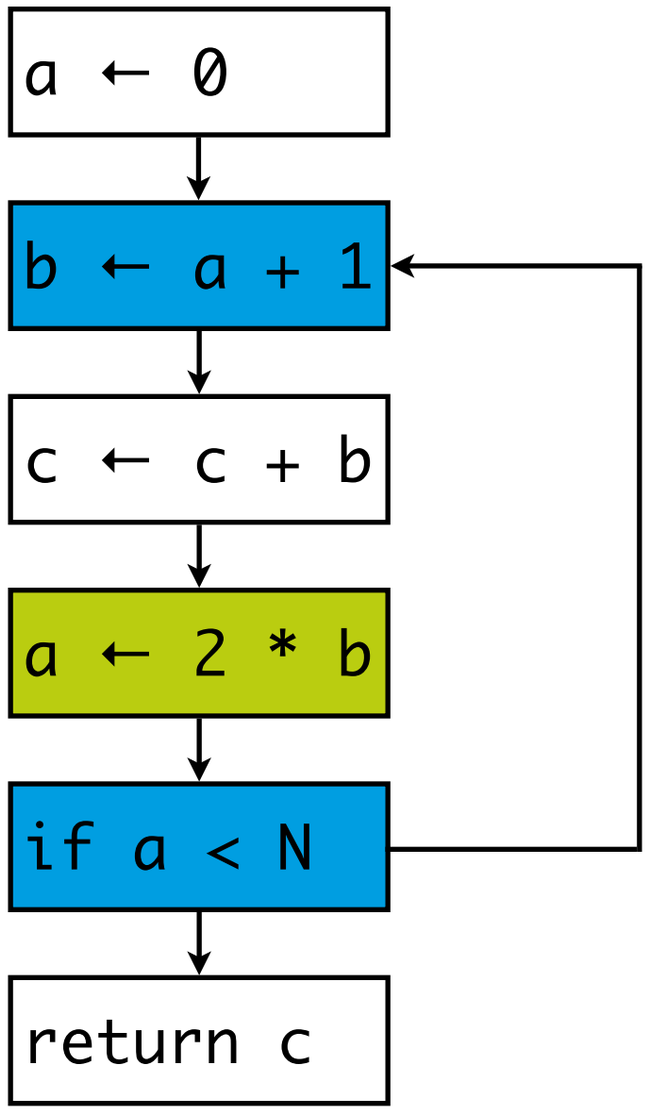
Recap: Liveness analysis

example



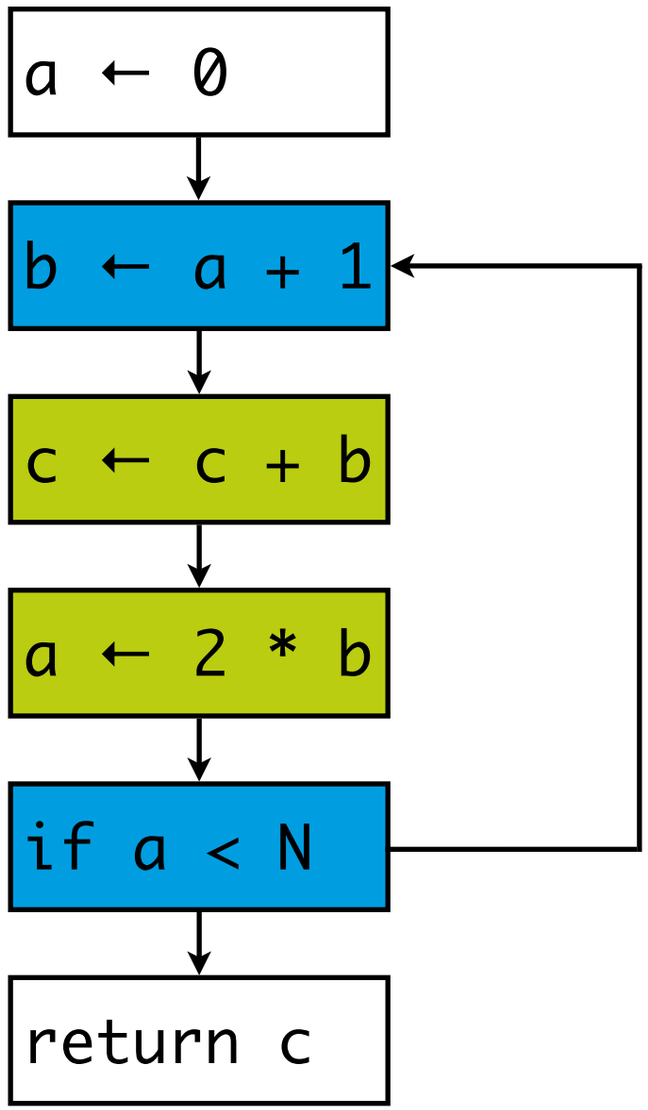
Recap: Liveness analysis

example



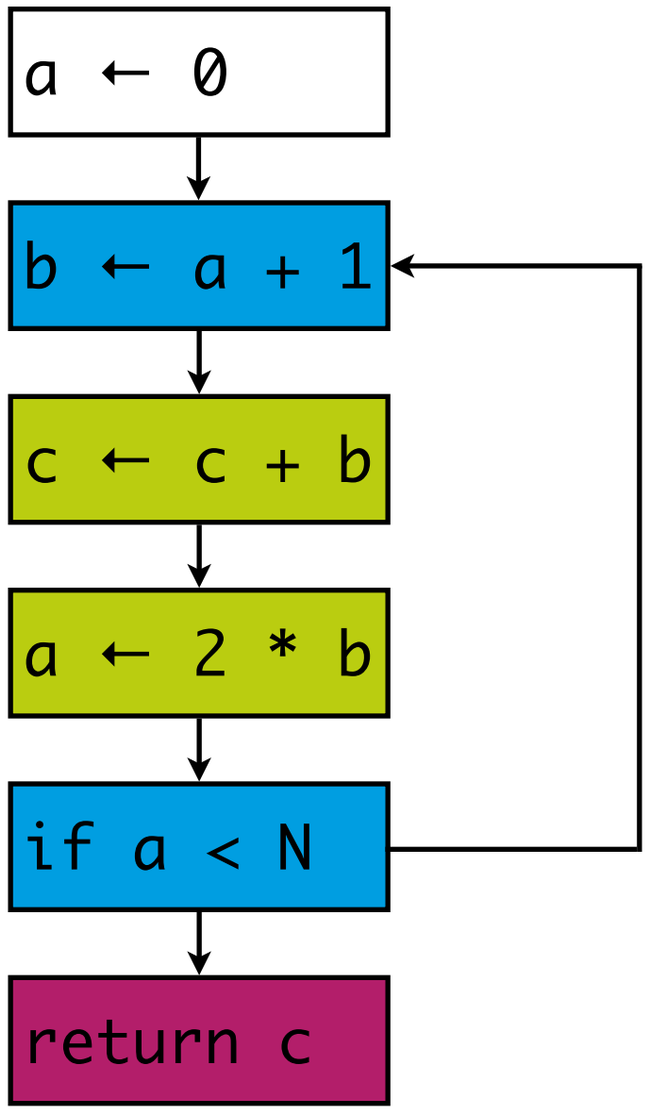
Recap: Liveness analysis

example



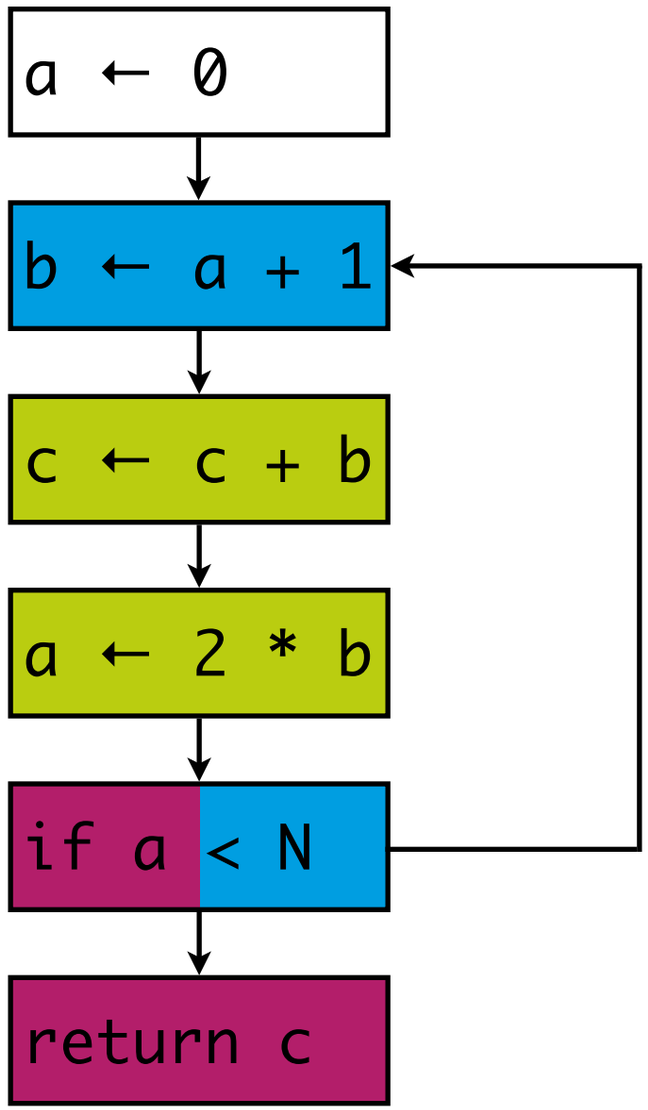
Recap: Liveness analysis

example



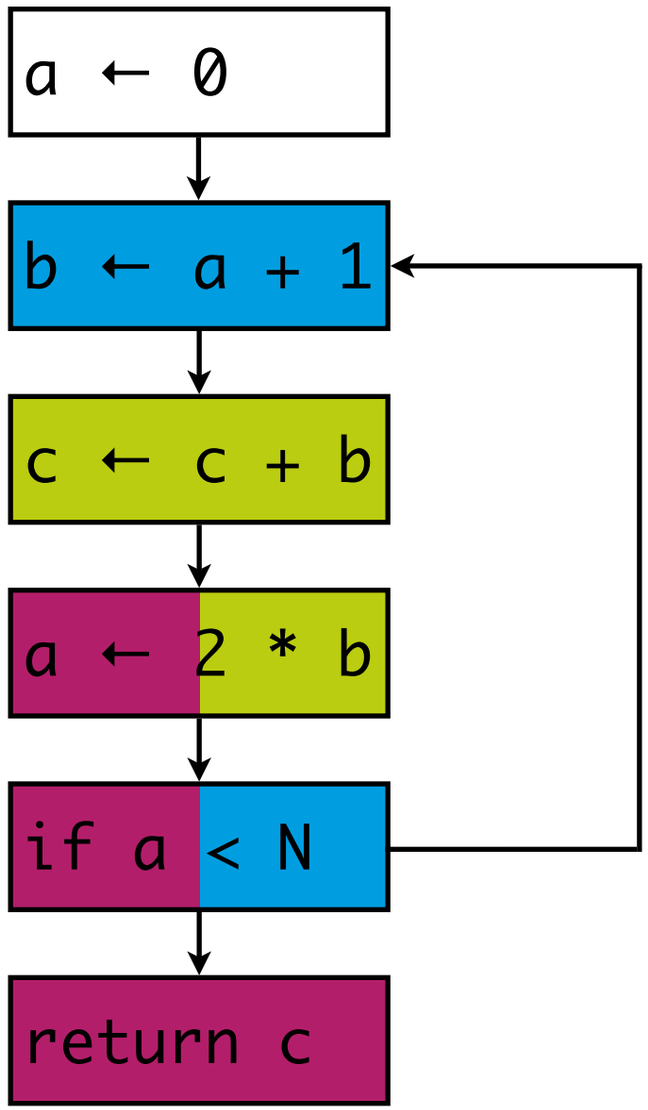
Recap: Liveness analysis

example



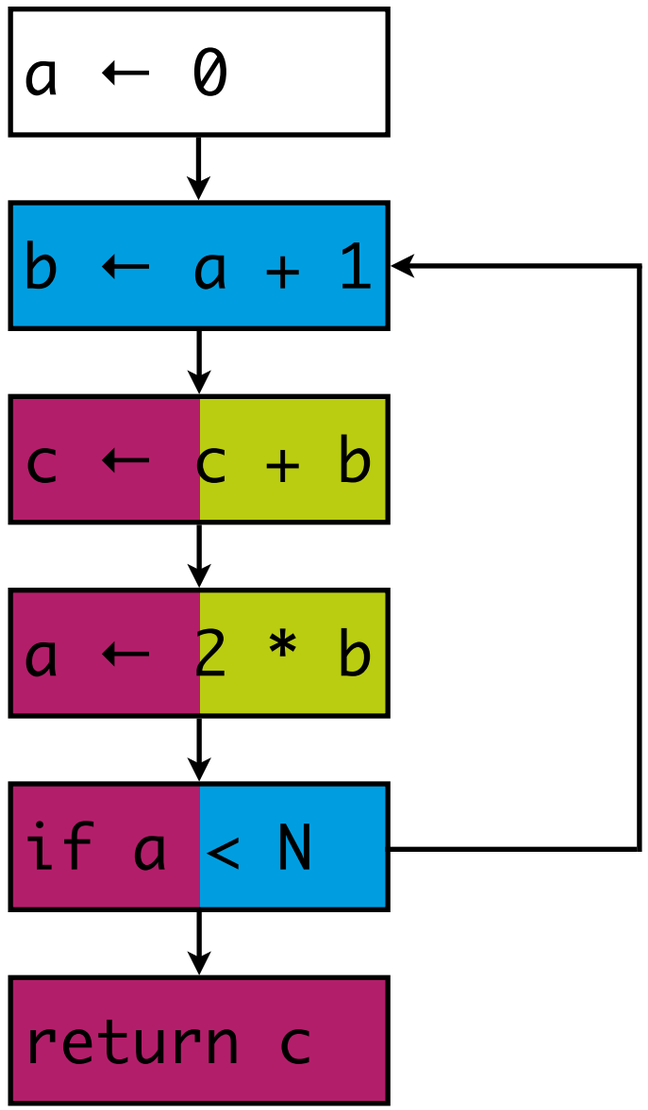
Recap: Liveness analysis

example



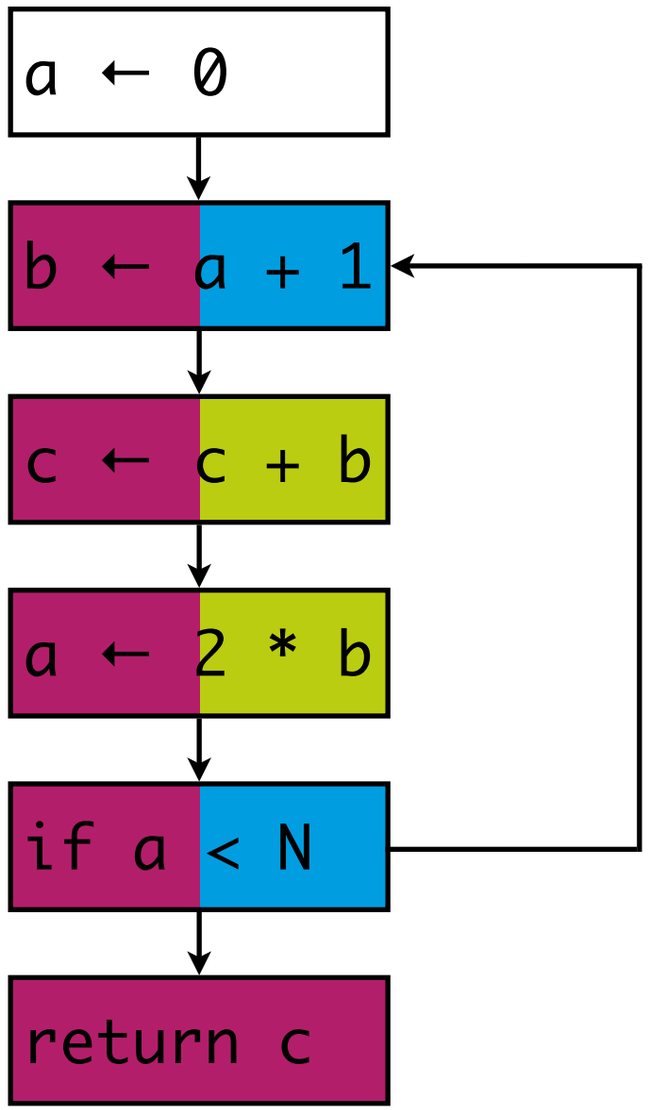
Recap: Liveness analysis

example



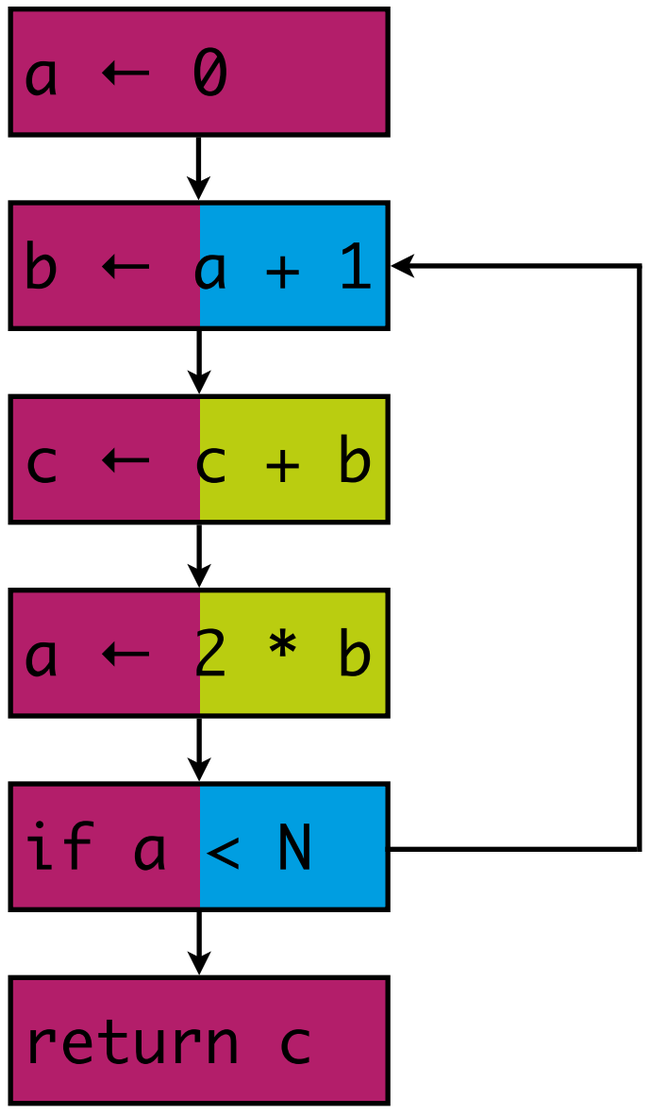
Recap: Liveness analysis

example



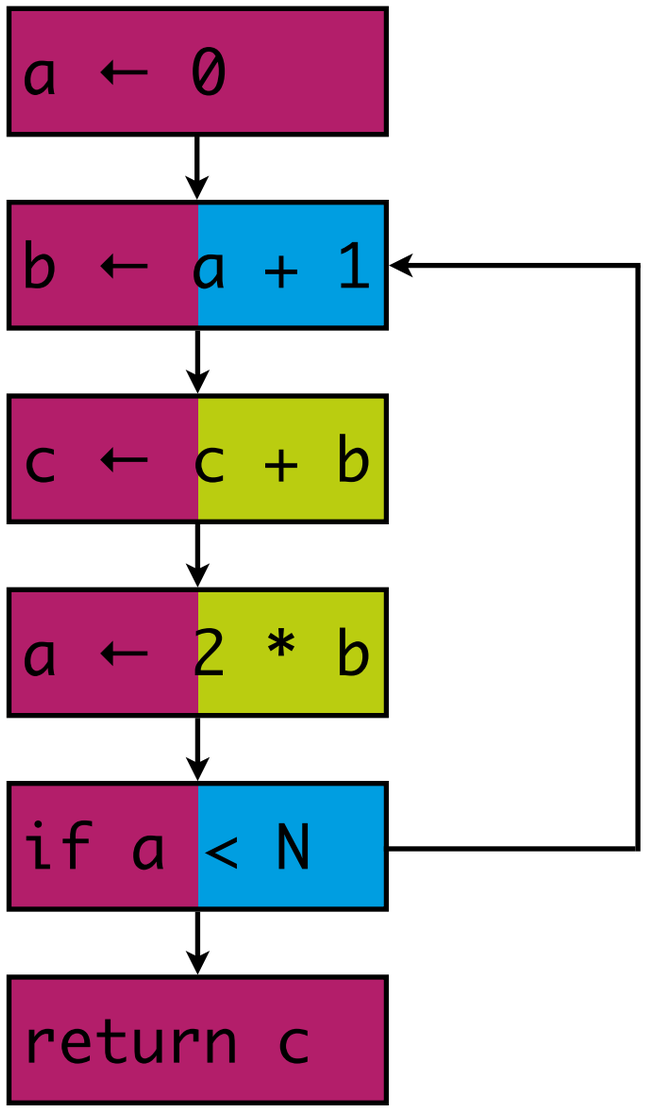
Recap: Liveness analysis

example



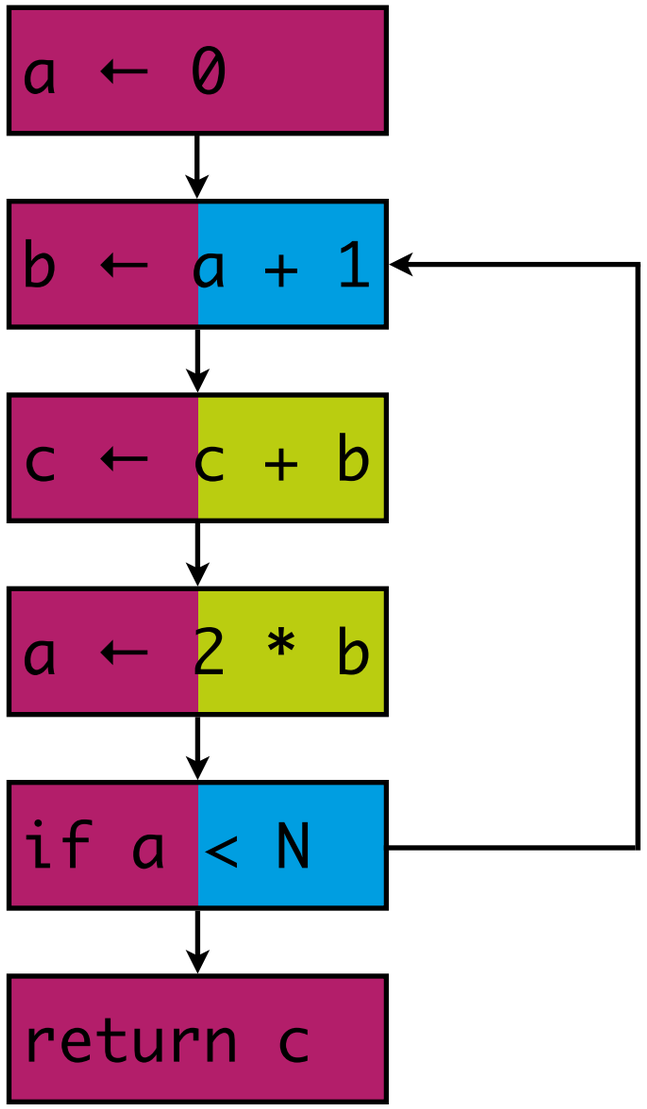
Interference graphs

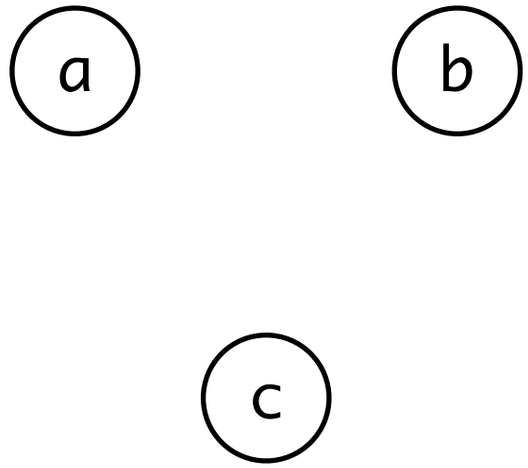
example



Interference graphs

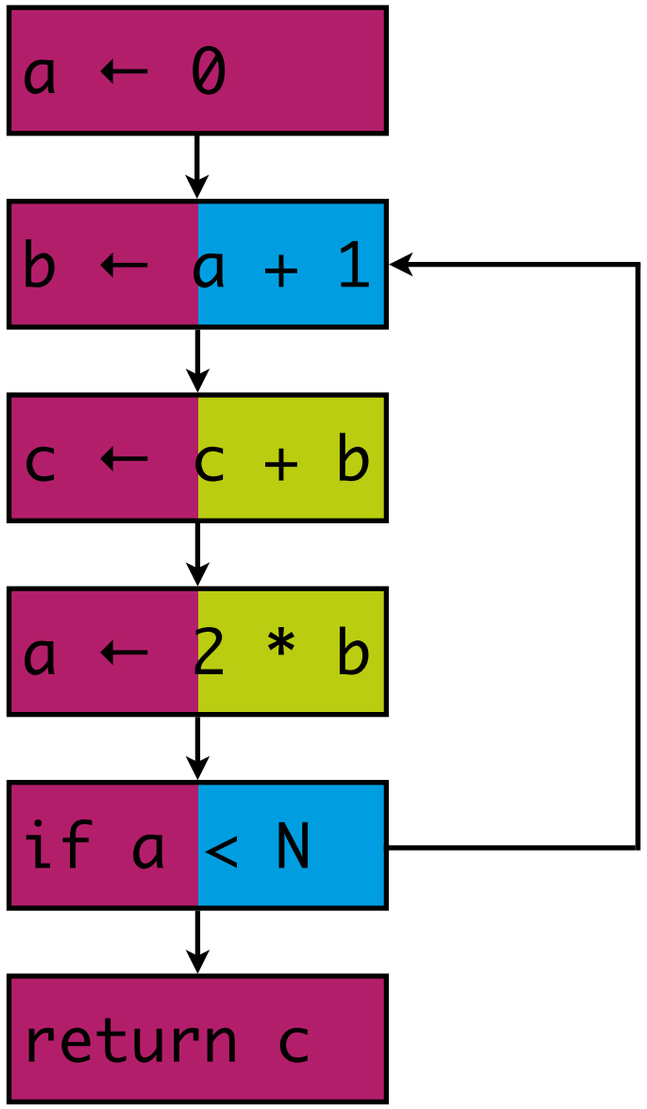
example

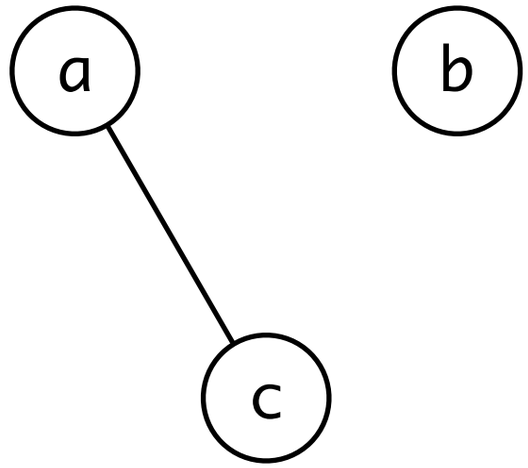




Interference graphs

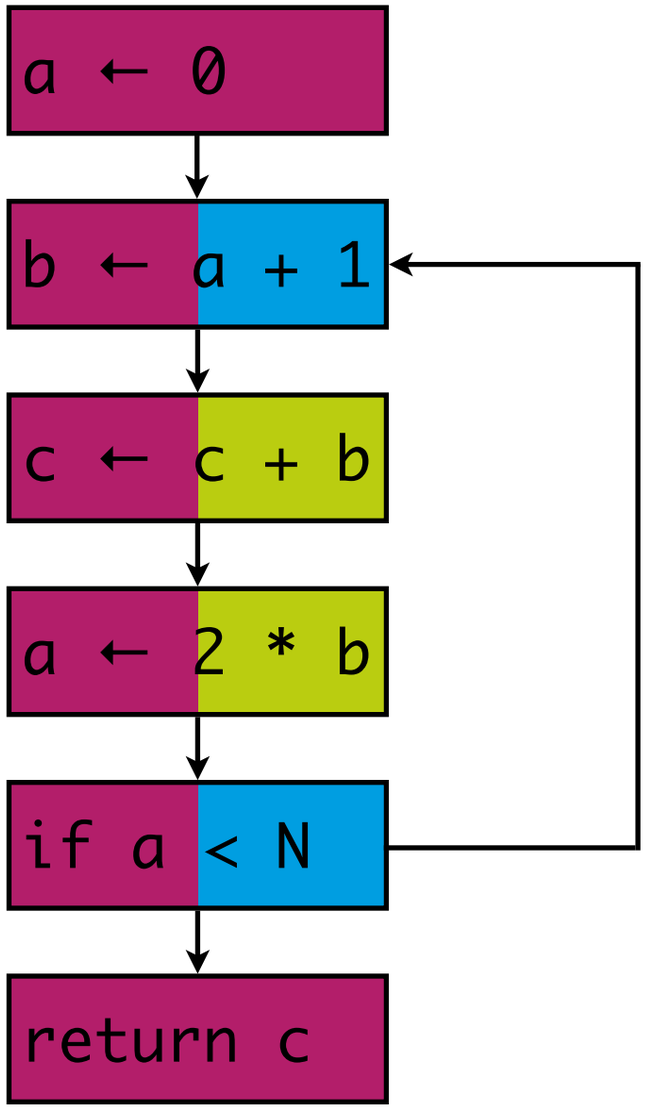
example

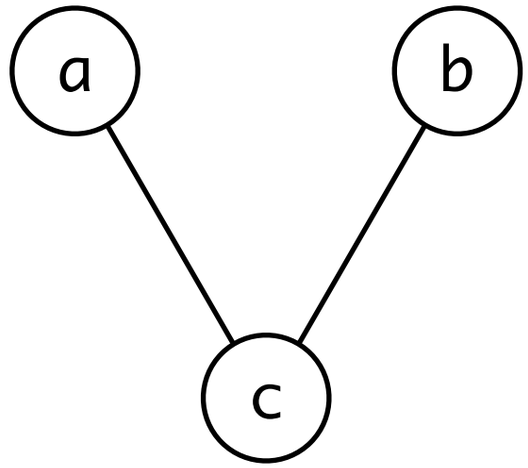




Interference graphs

example



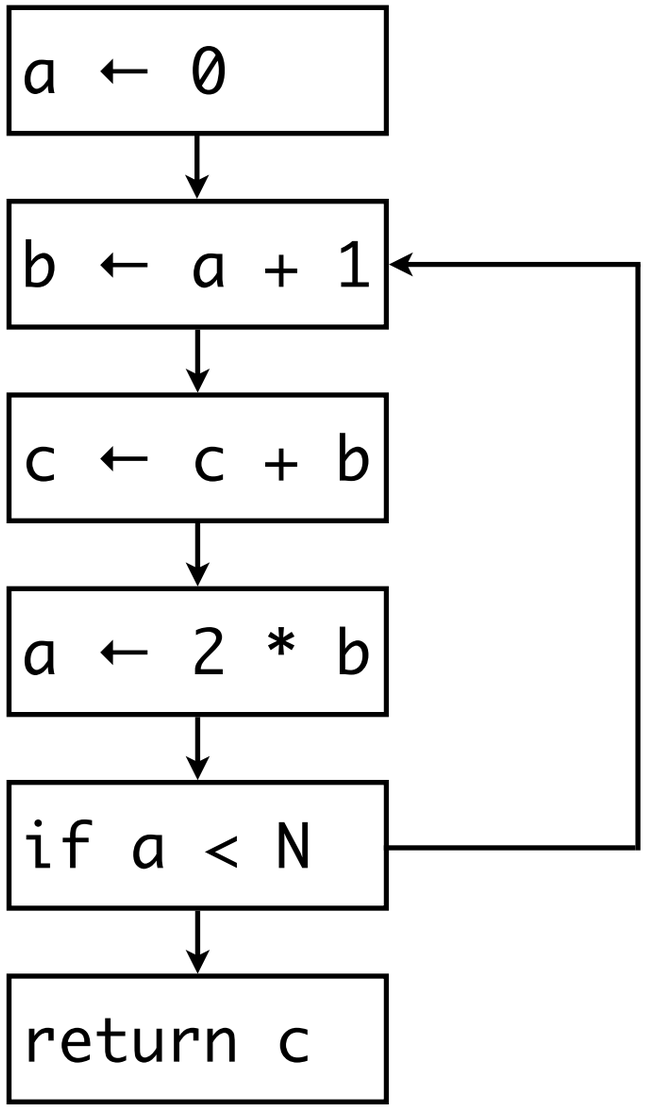


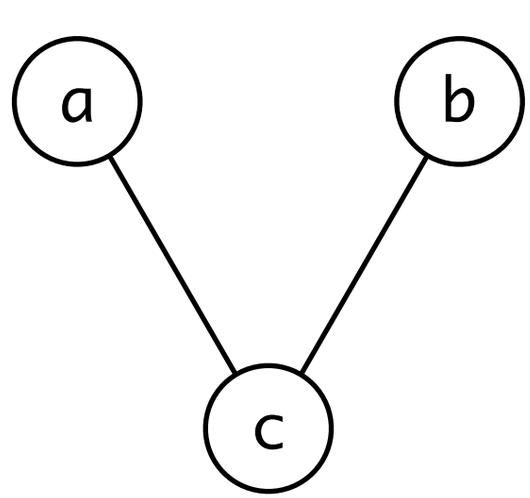
II

graph colouring

Graph colouring

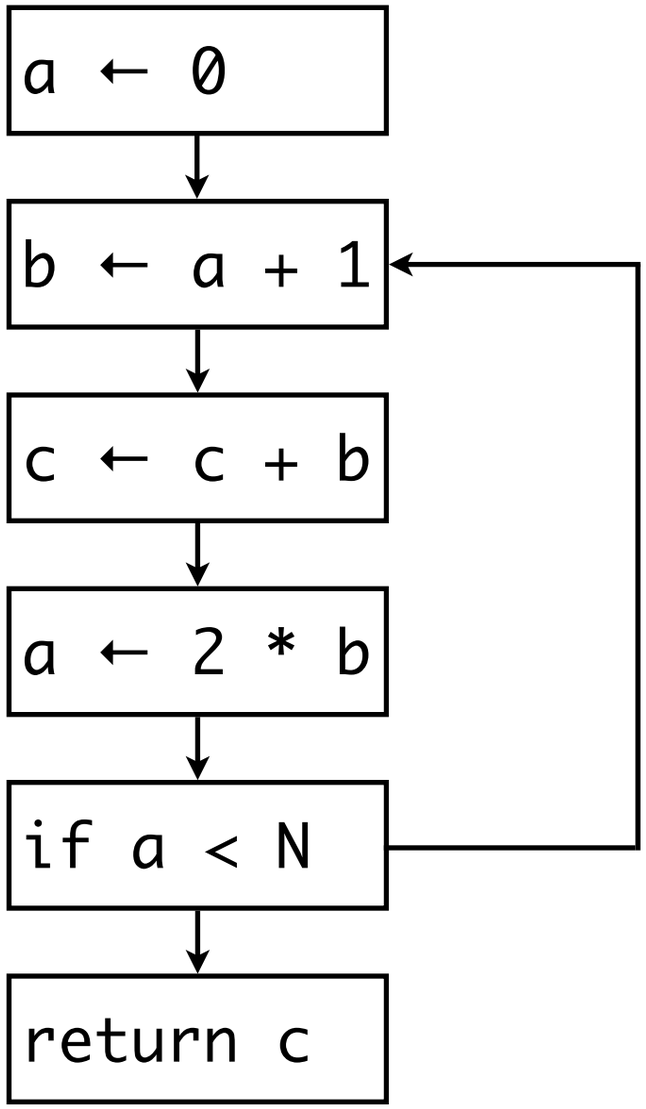
example





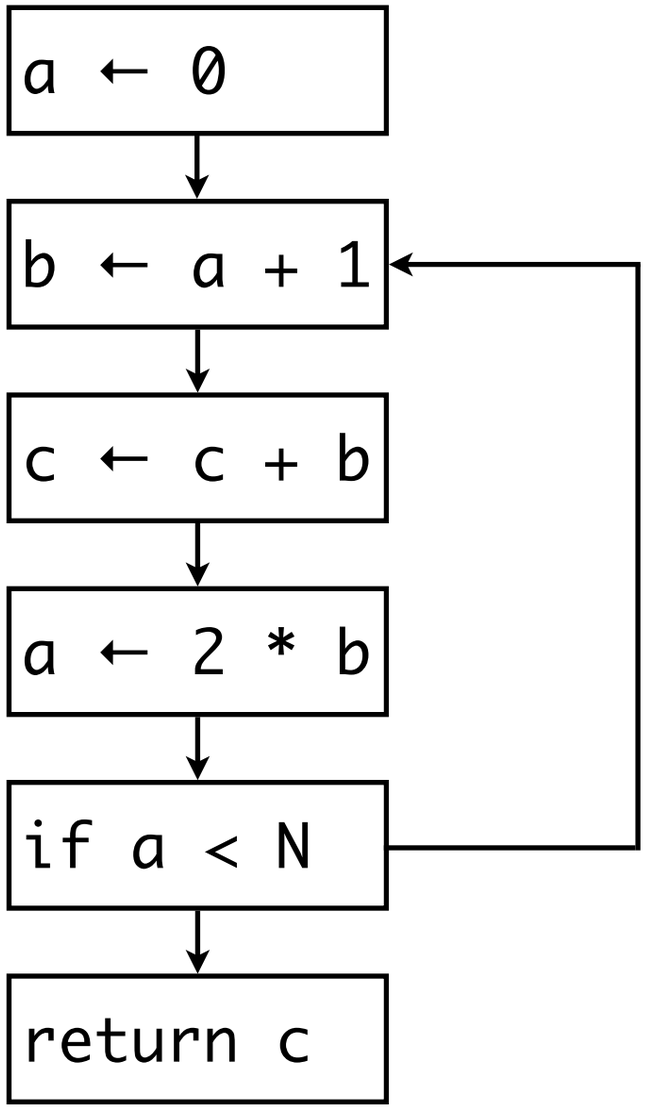
Graph colouring

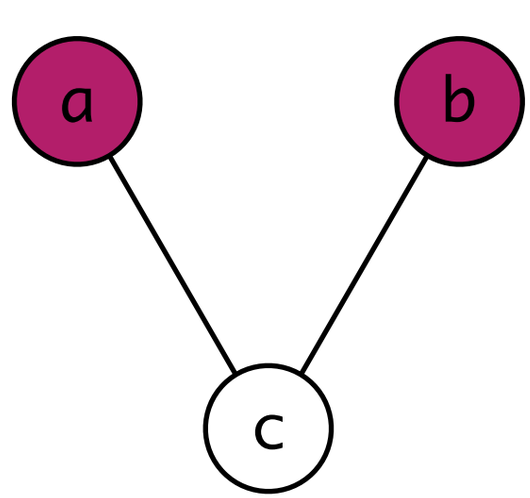
example



Graph colouring

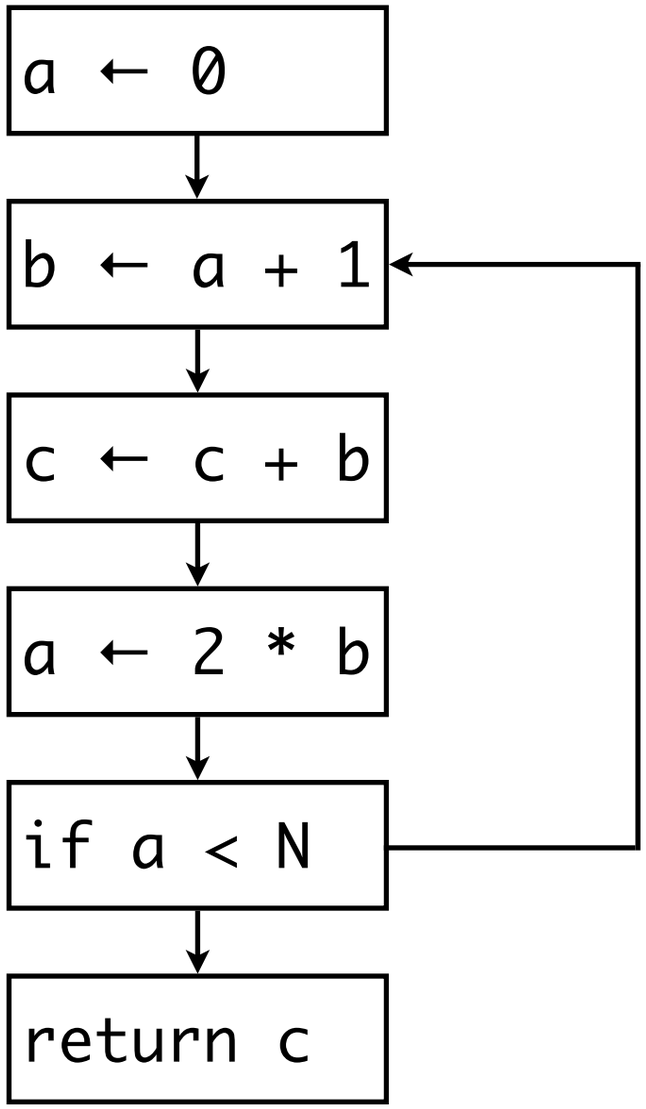
example

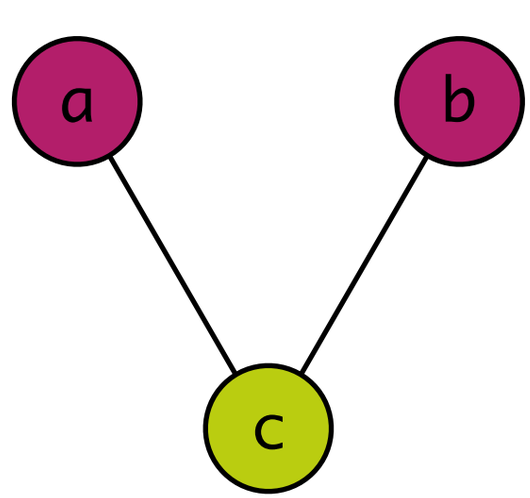




Graph colouring

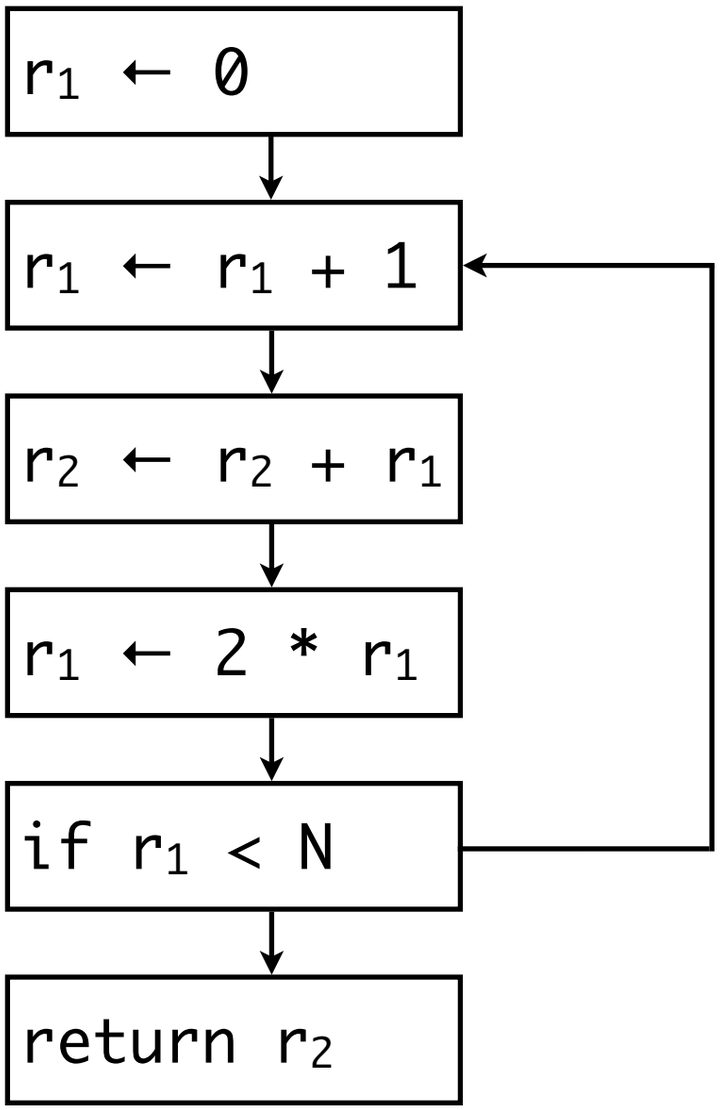
example

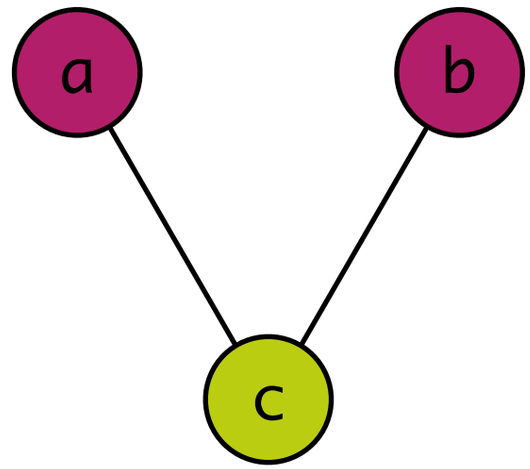




Graph colouring

example





Graph colouring

steps

simplify

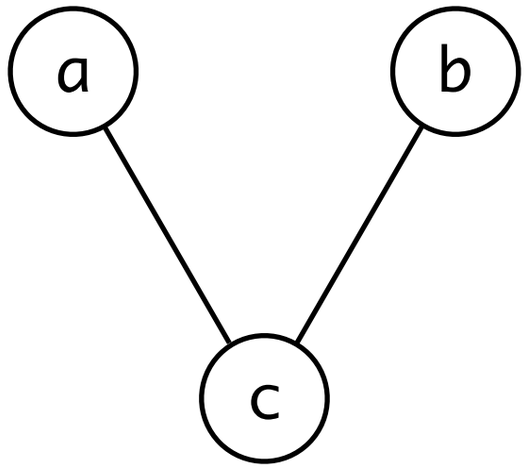
remove node of insignificant degree (fewer than k edges)

select

add node, select colour

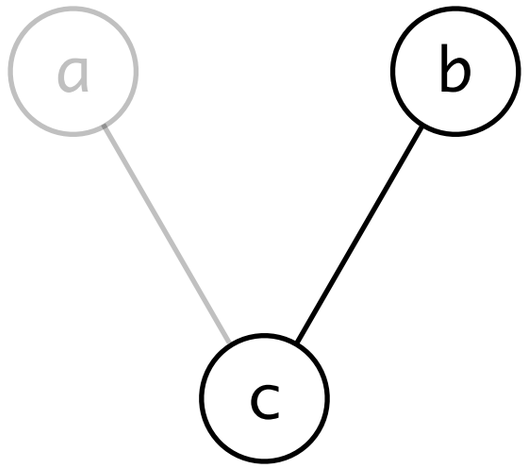
Graph colouring

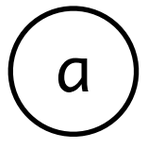
example with 2 colours



Graph colouring

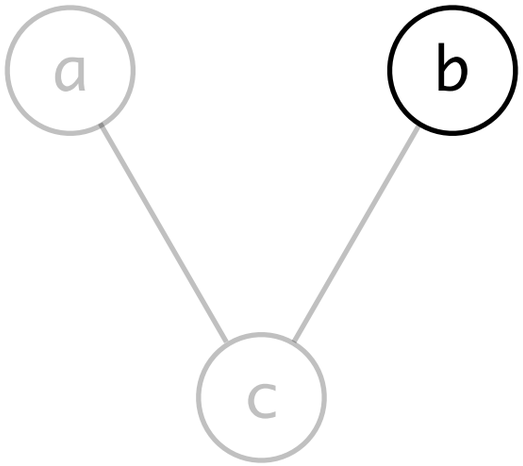
example with 2 colours

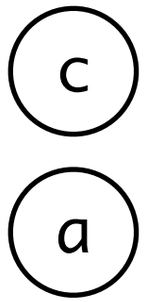




Graph colouring

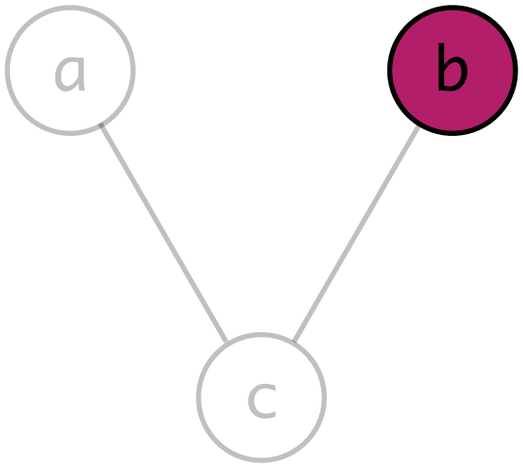
example with 2 colours

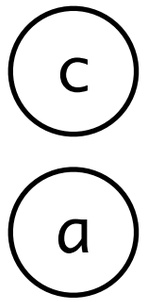




Graph colouring

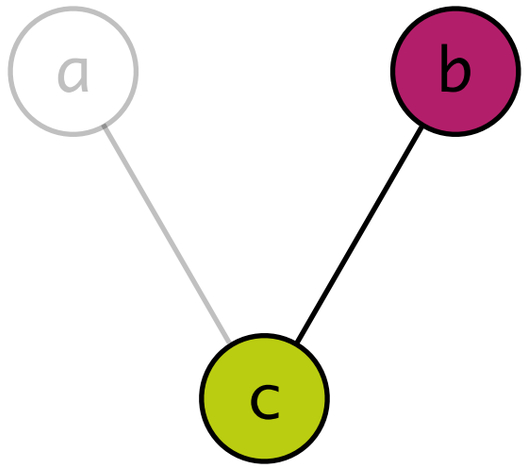
example with 2 colours

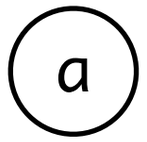




Graph colouring

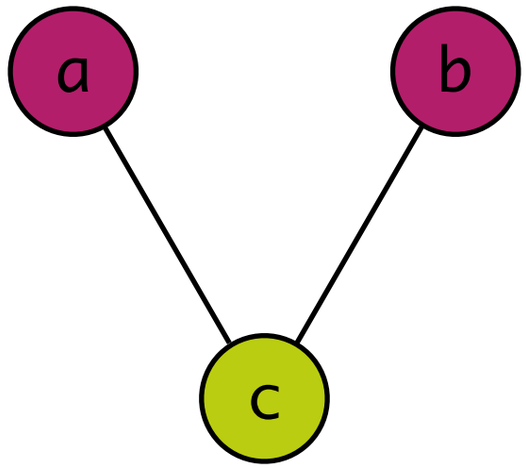
example with 2 colours





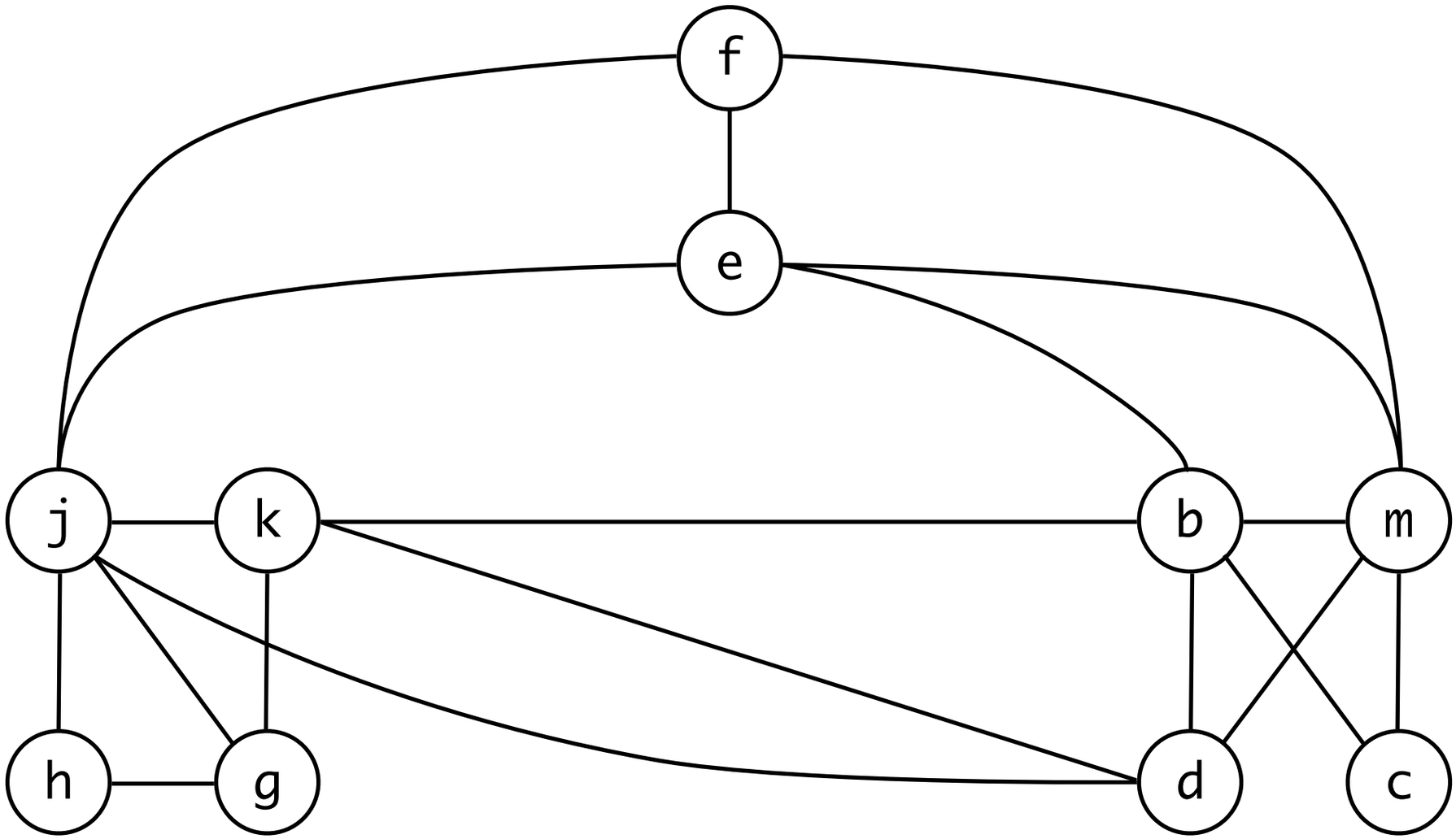
Graph colouring

example with 2 colours



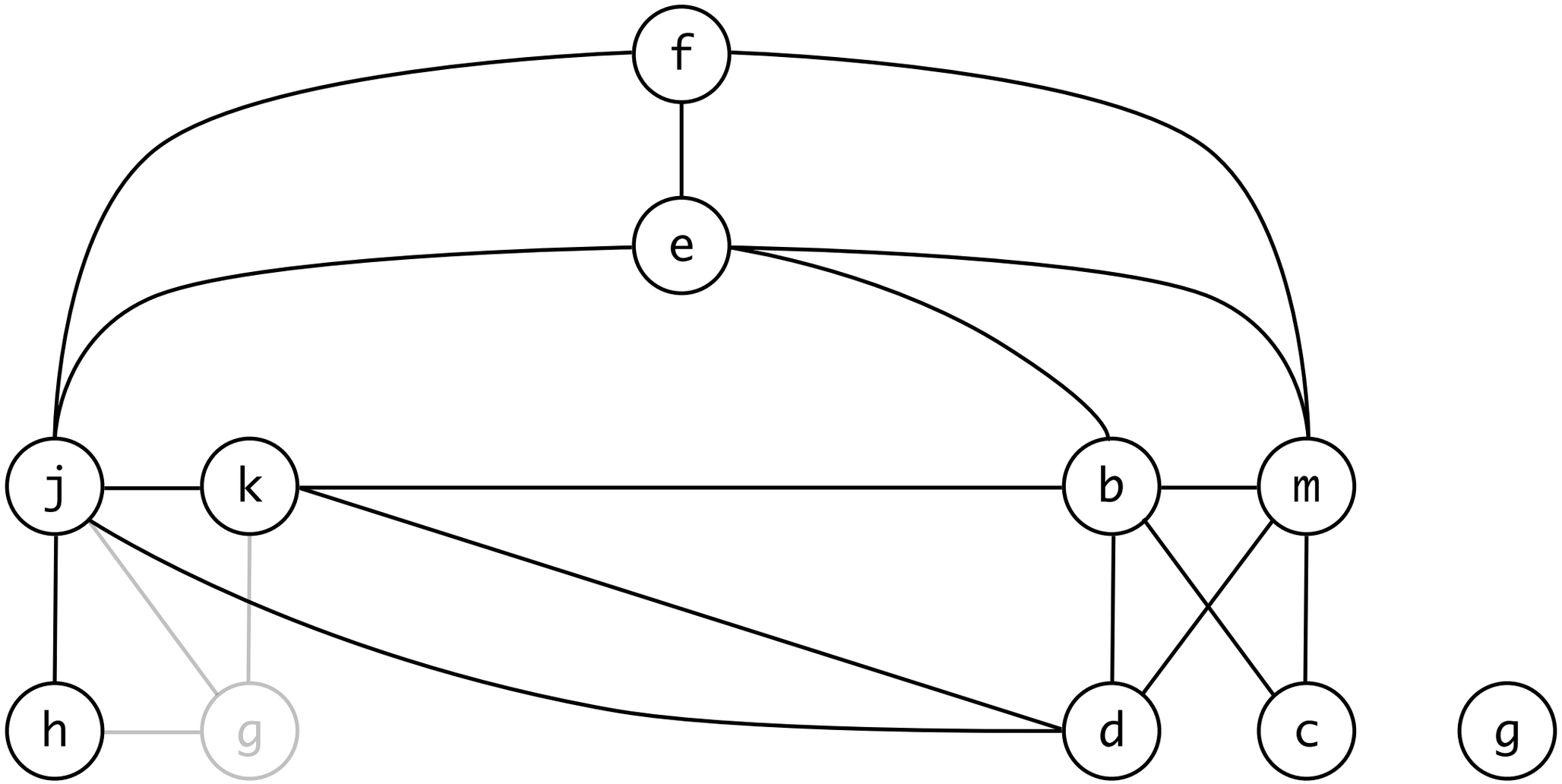
Graph colouring

example with 4 colours



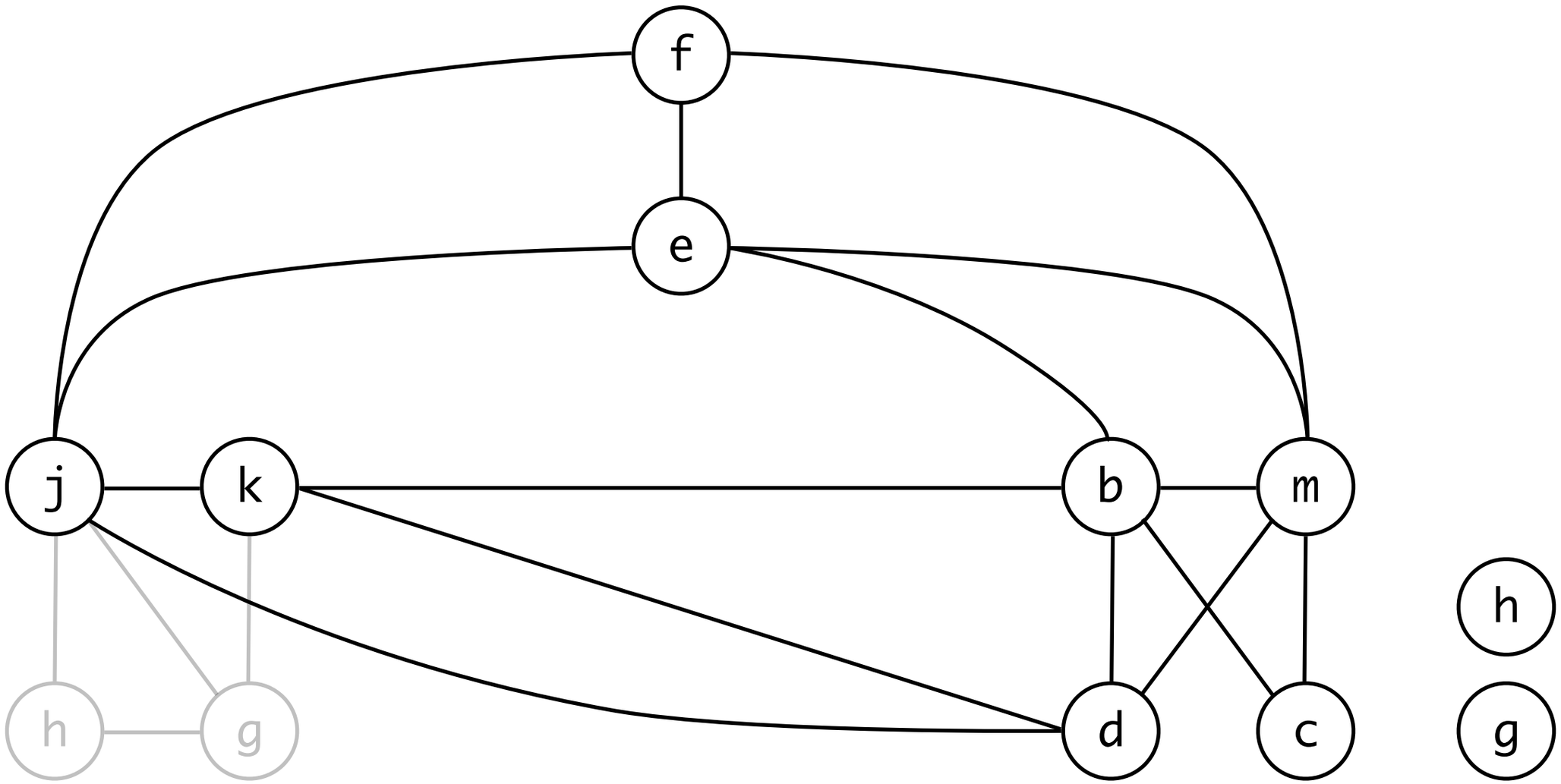
Graph colouring

example with 4 colours



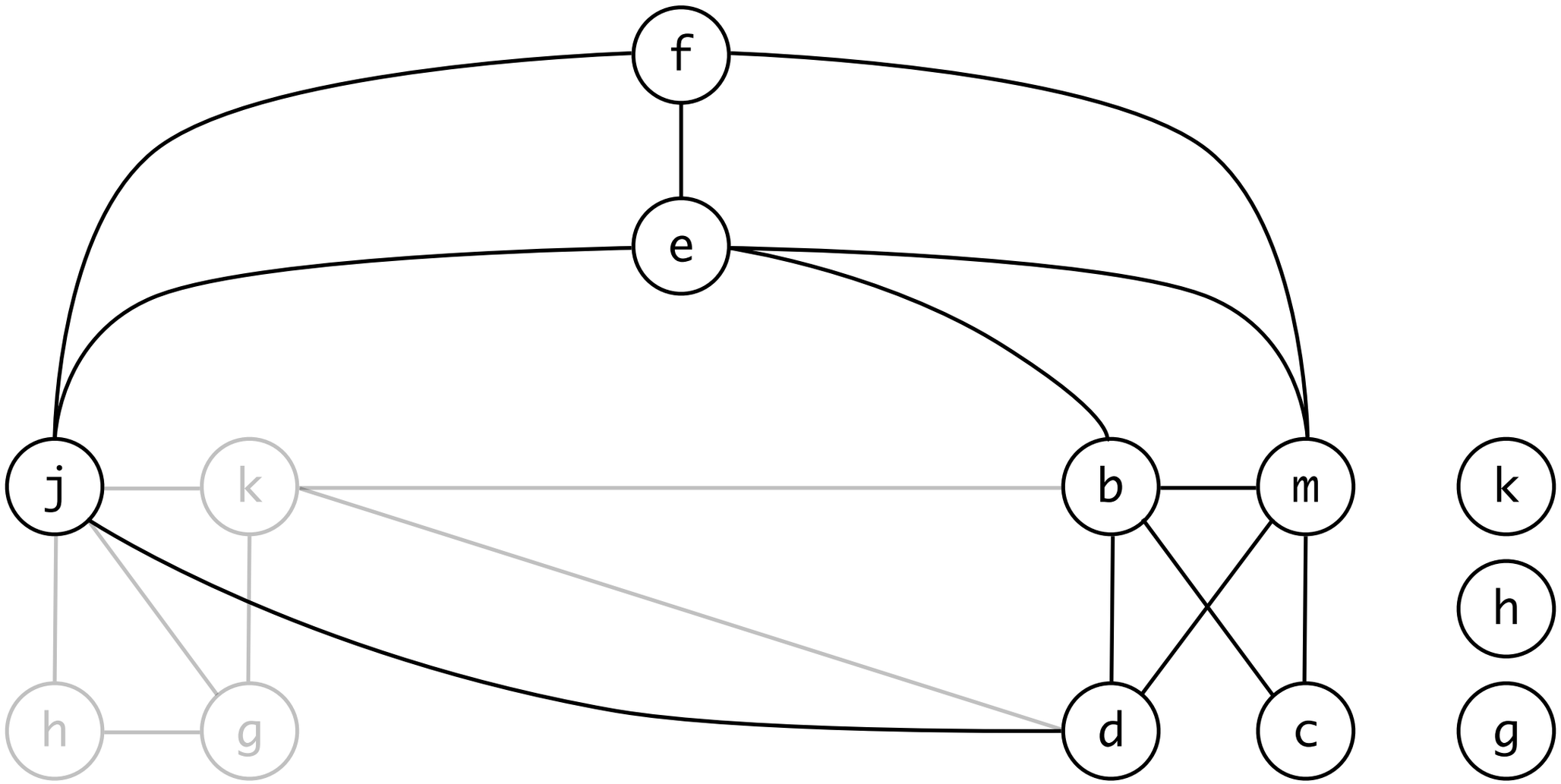
Graph colouring

example with 4 colours



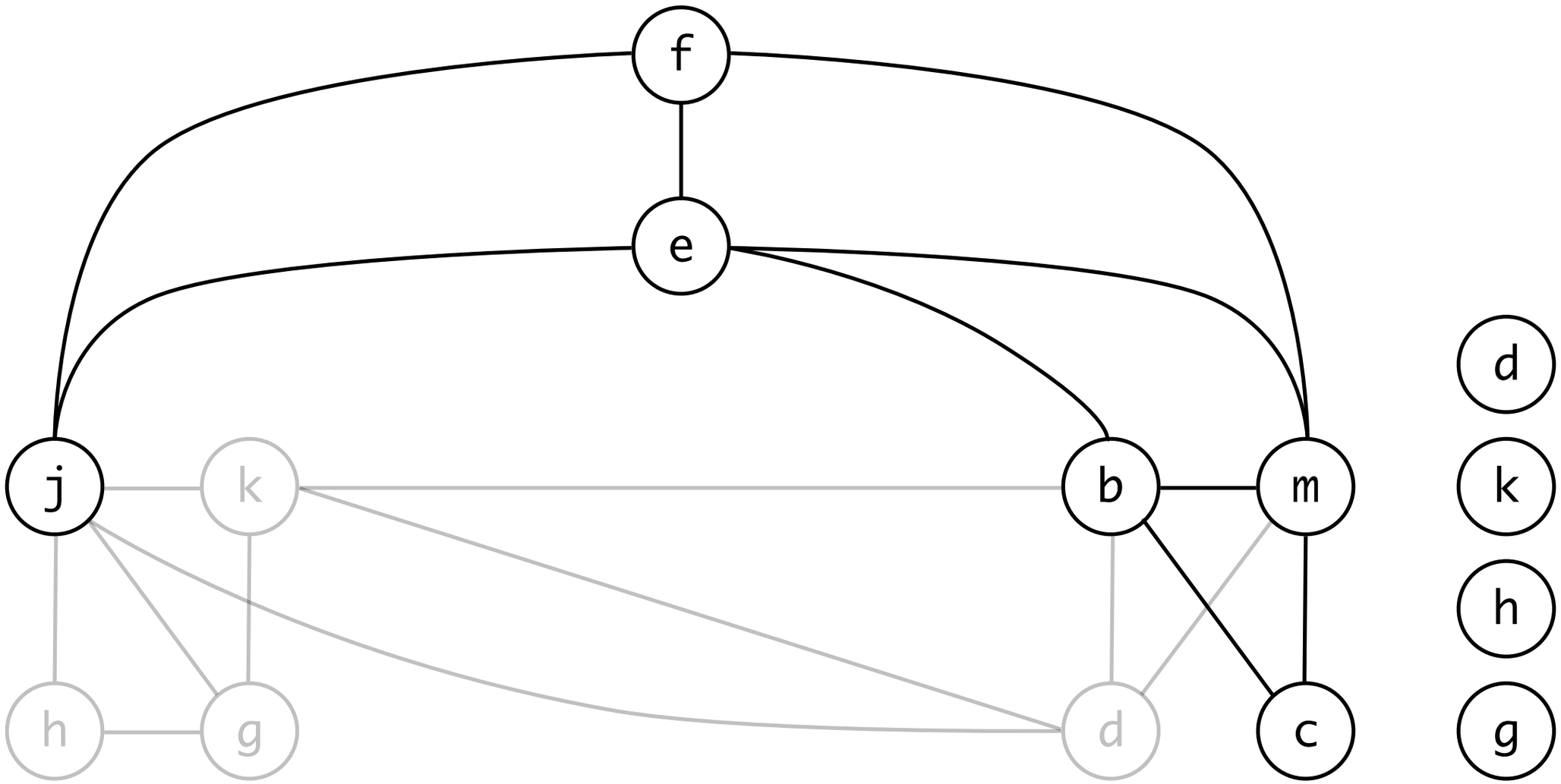
Graph colouring

example with 4 colours



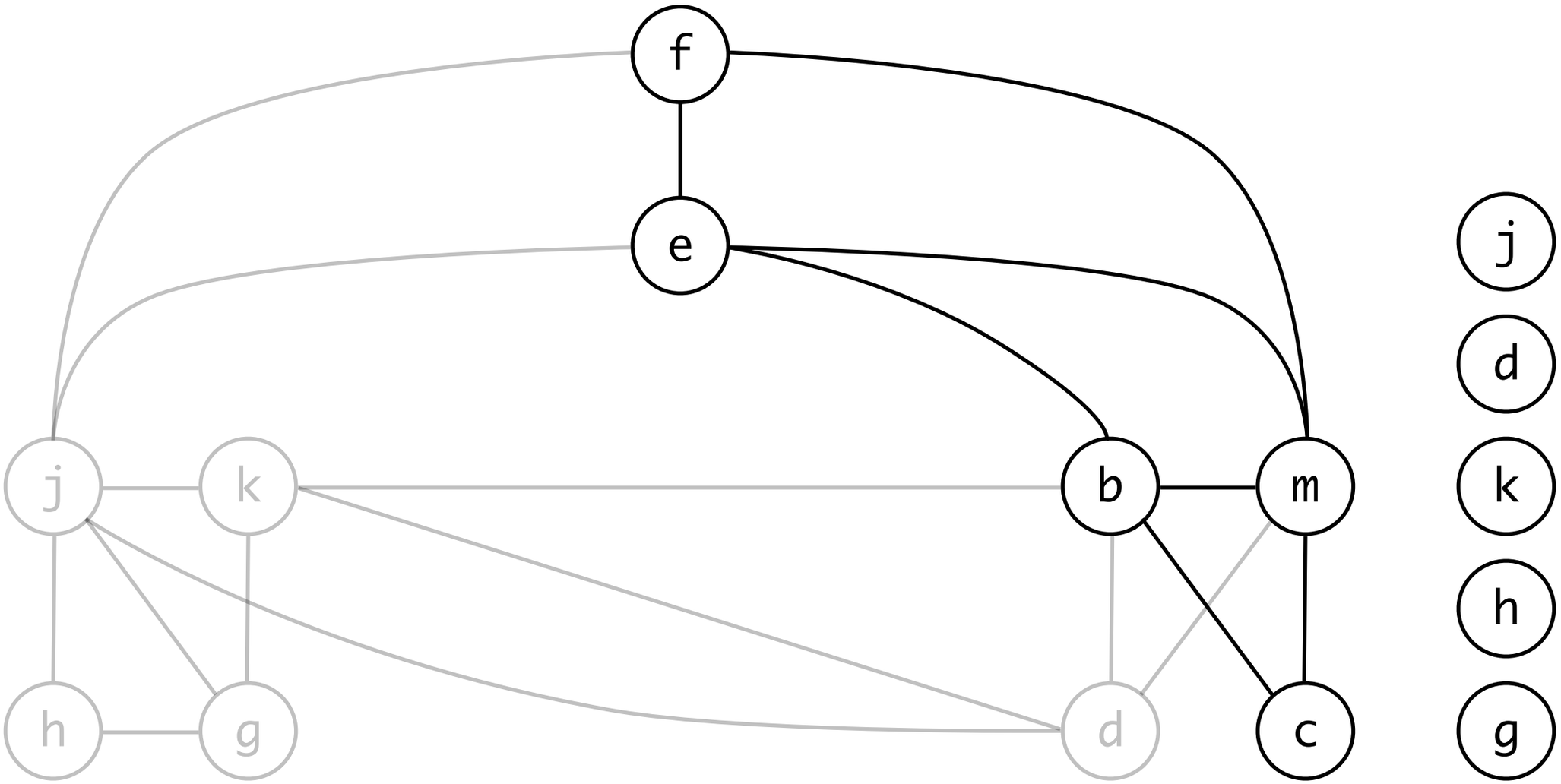
Graph colouring

example with 4 colours



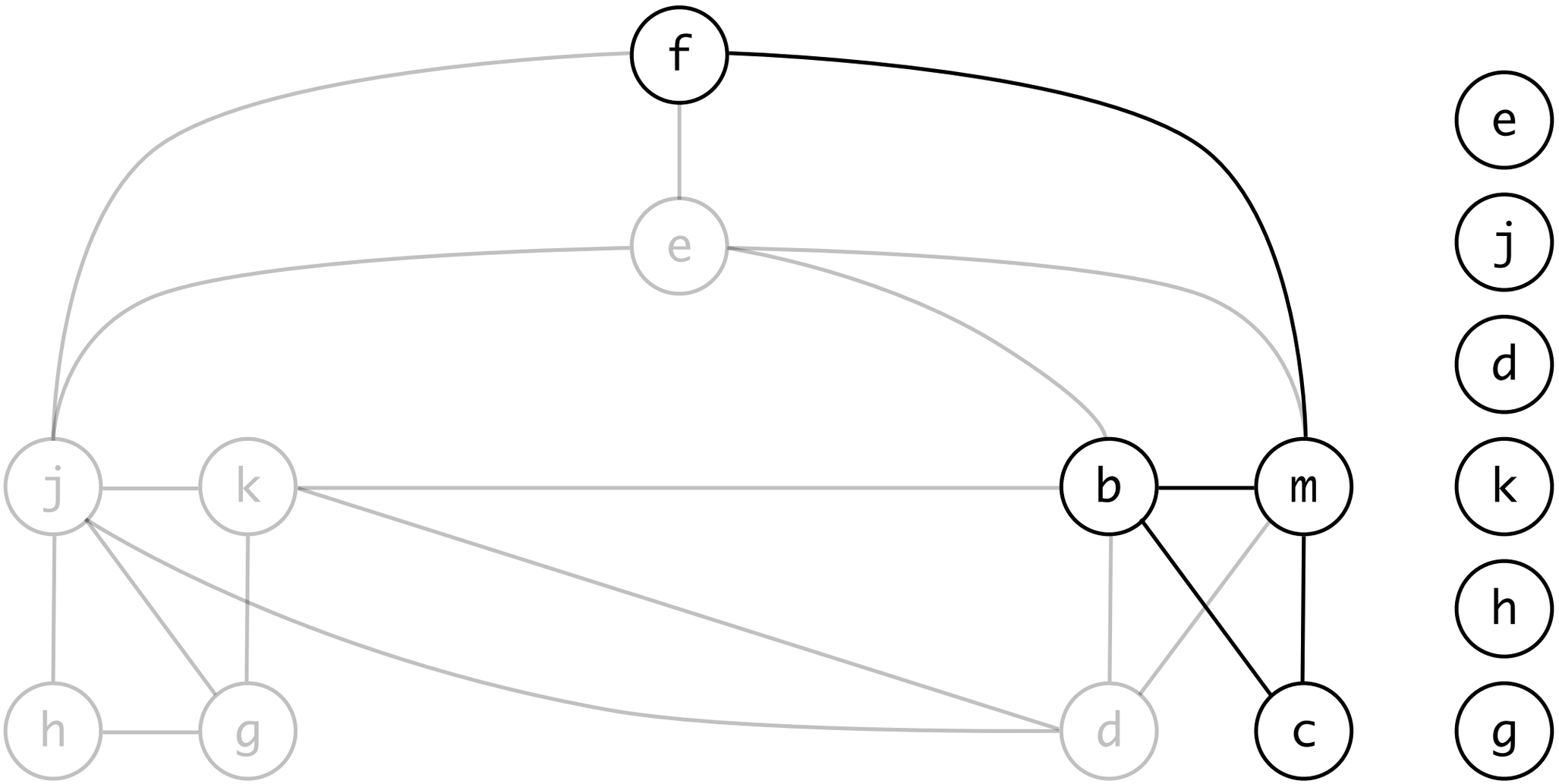
Graph colouring

example with 4 colours



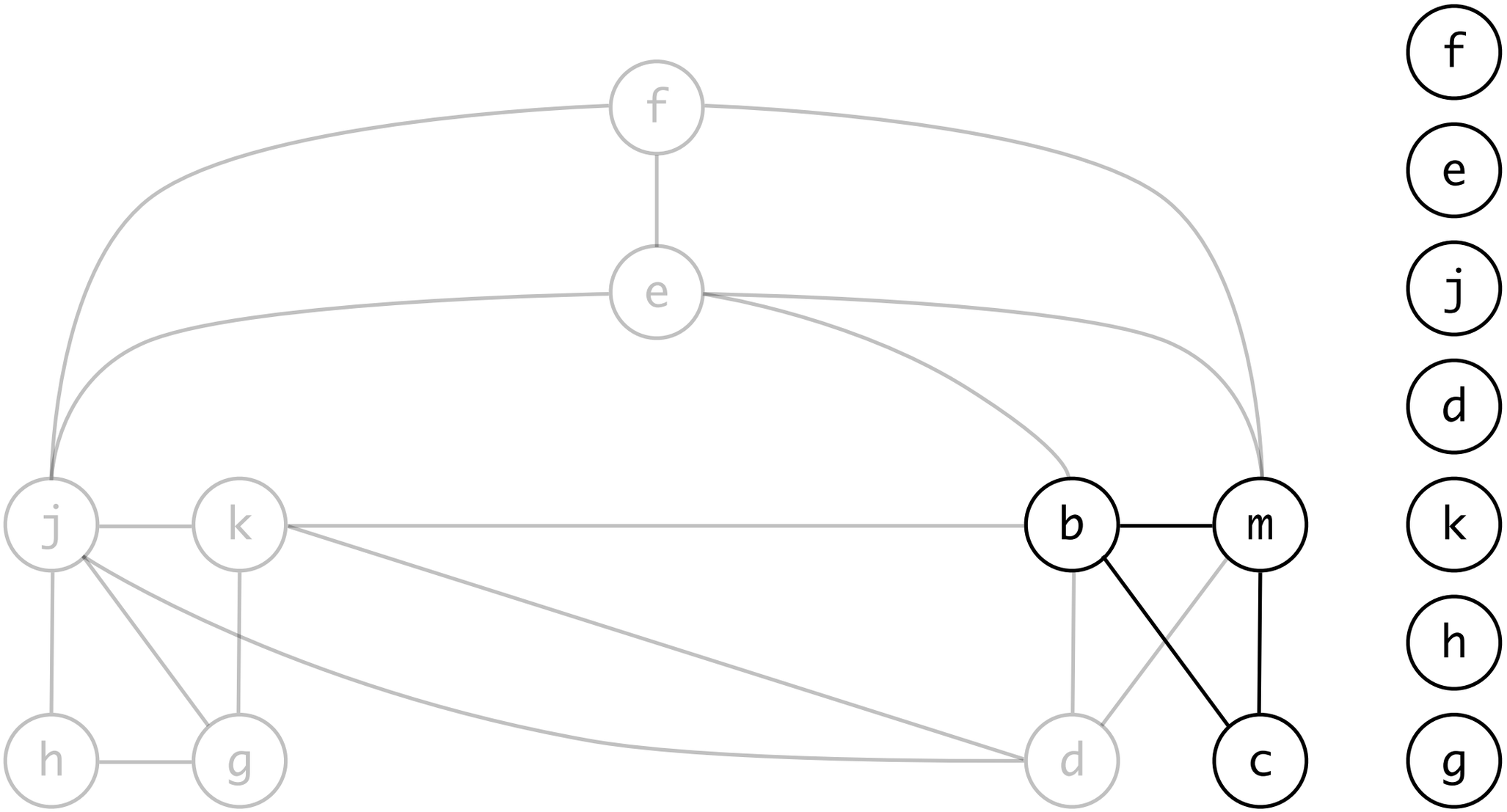
Graph colouring

example with 4 colours



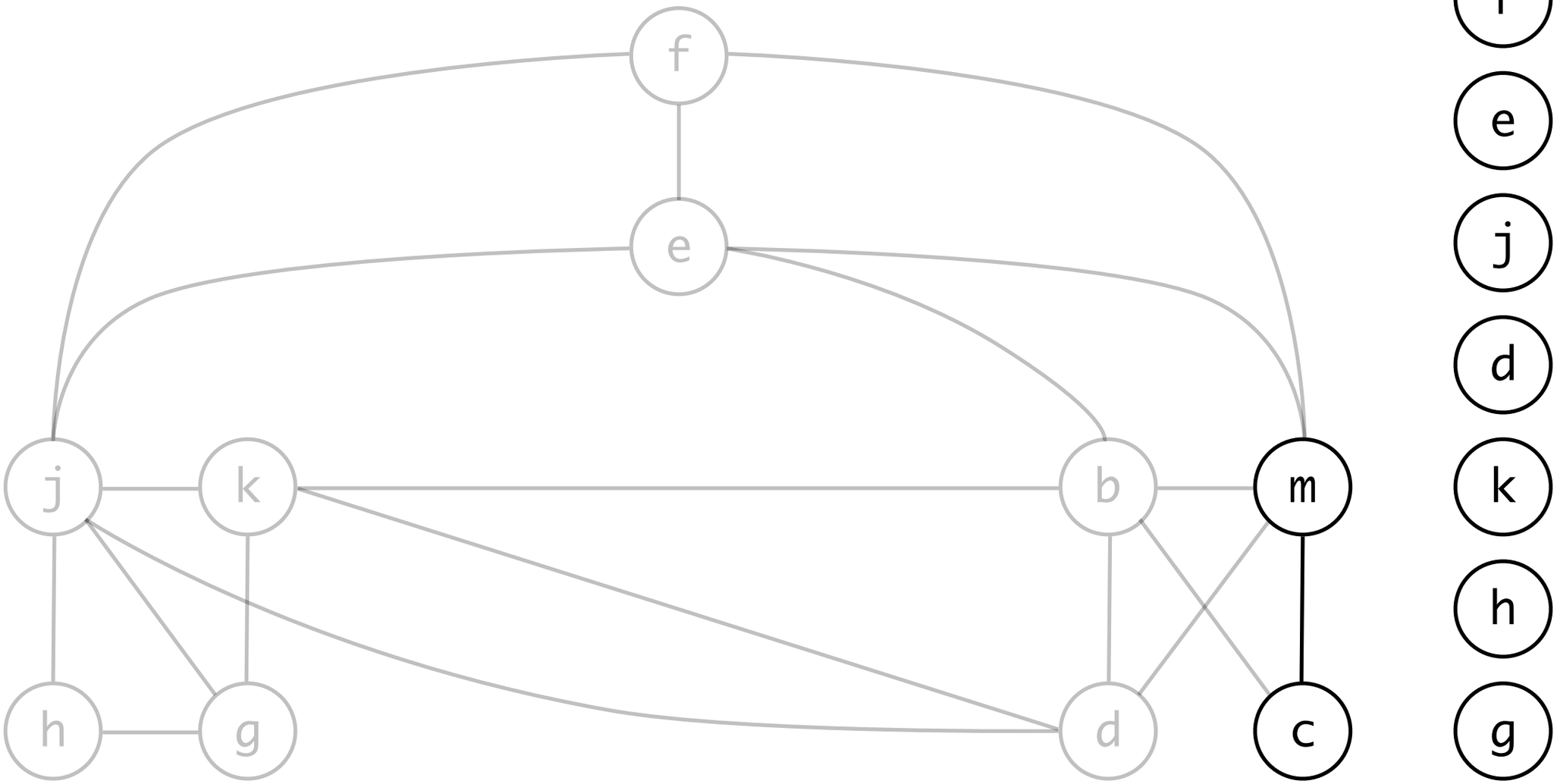
Graph colouring

example with 4 colours



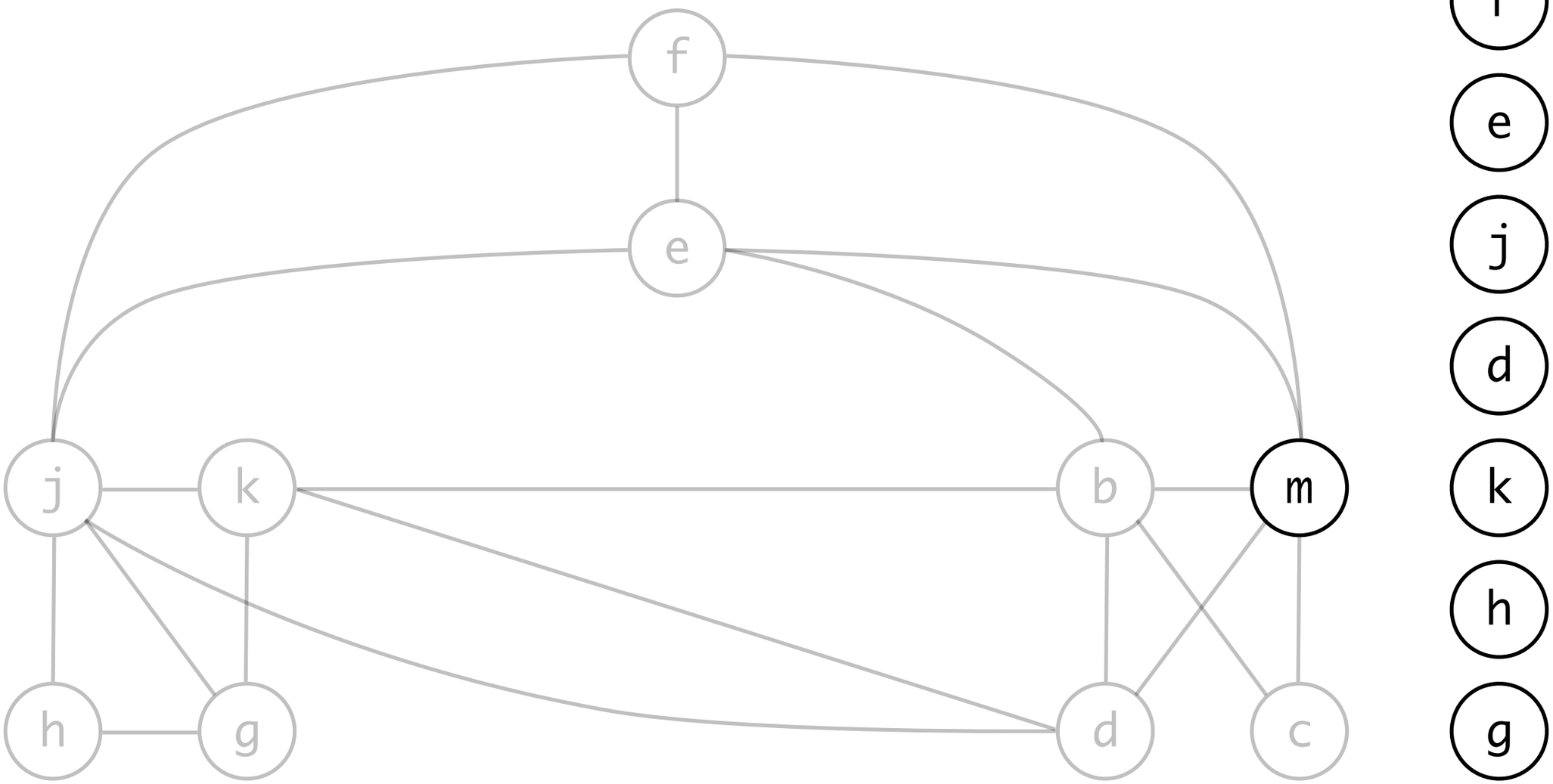
Graph colouring

example with 4 colours



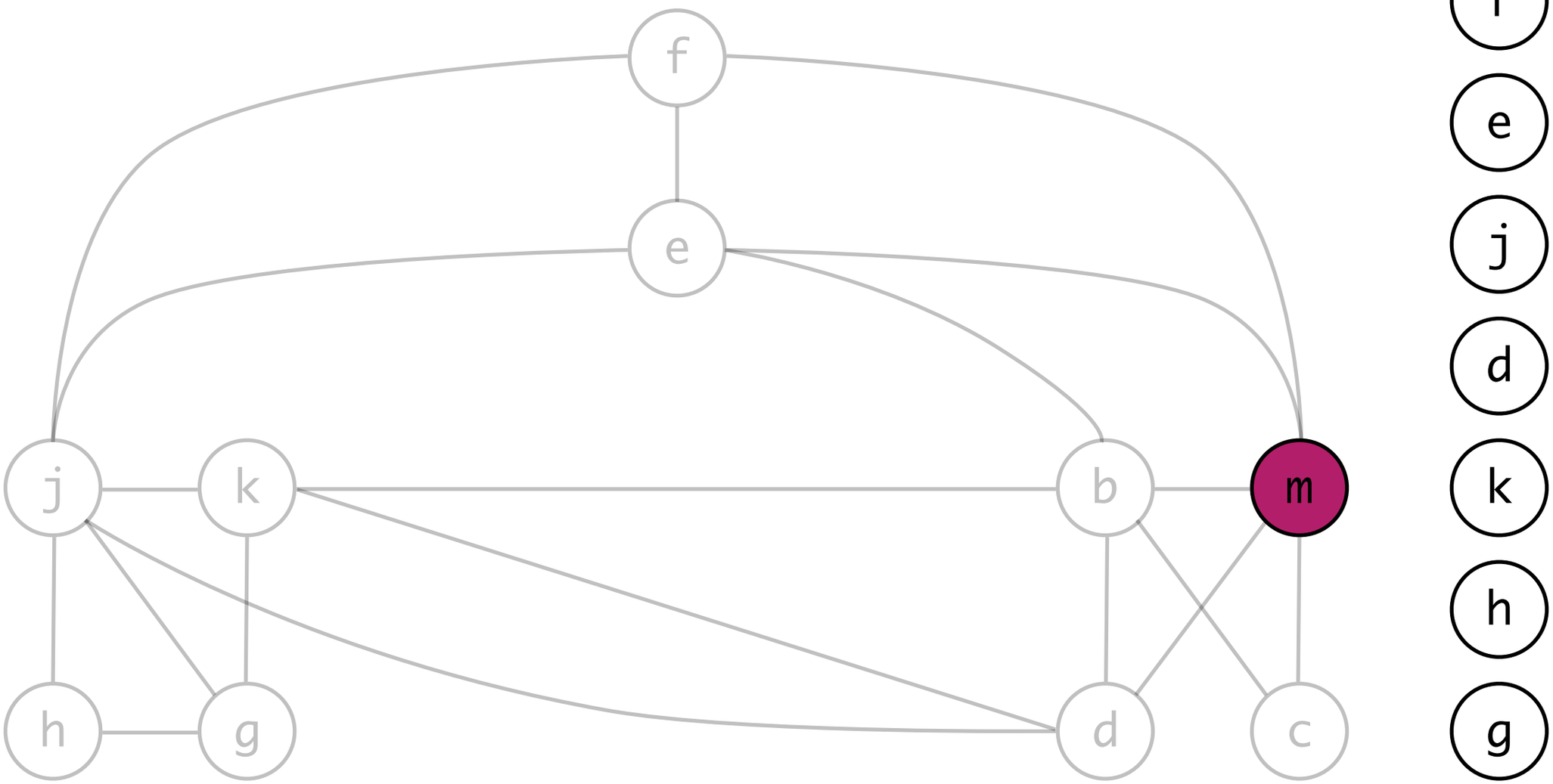
Graph colouring

example with 4 colours



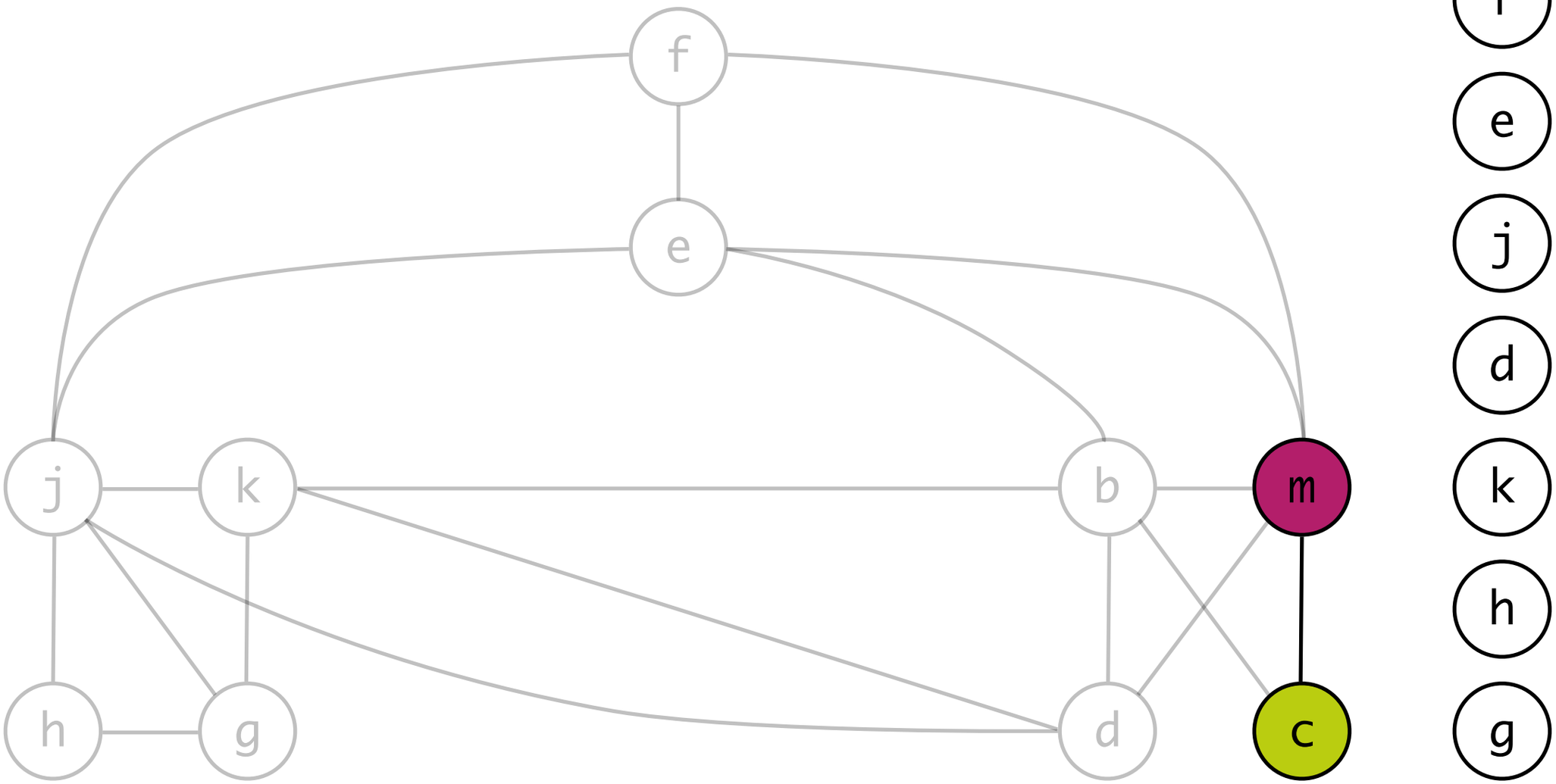
Graph colouring

example with 4 colours



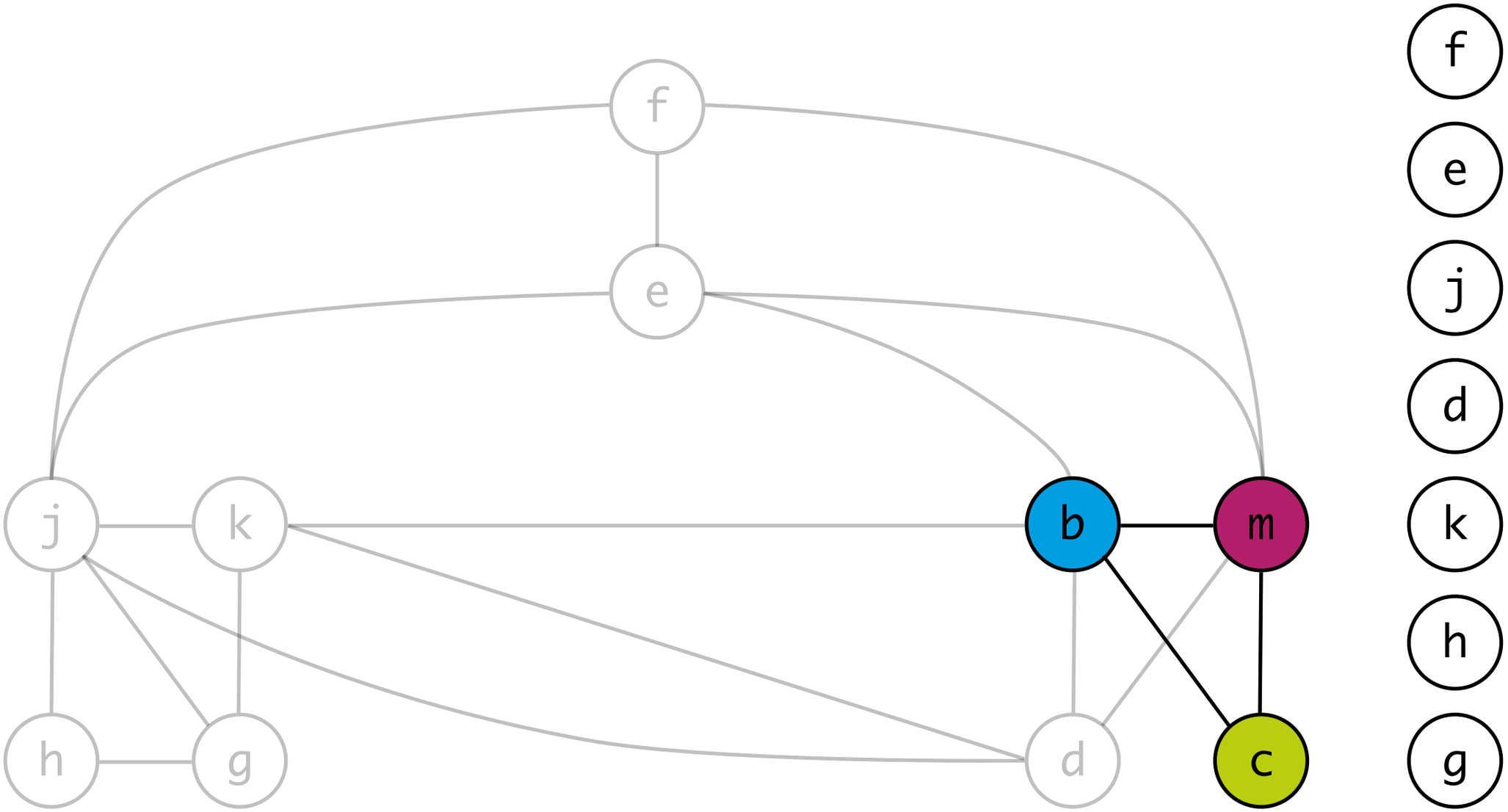
Graph colouring

example with 4 colours



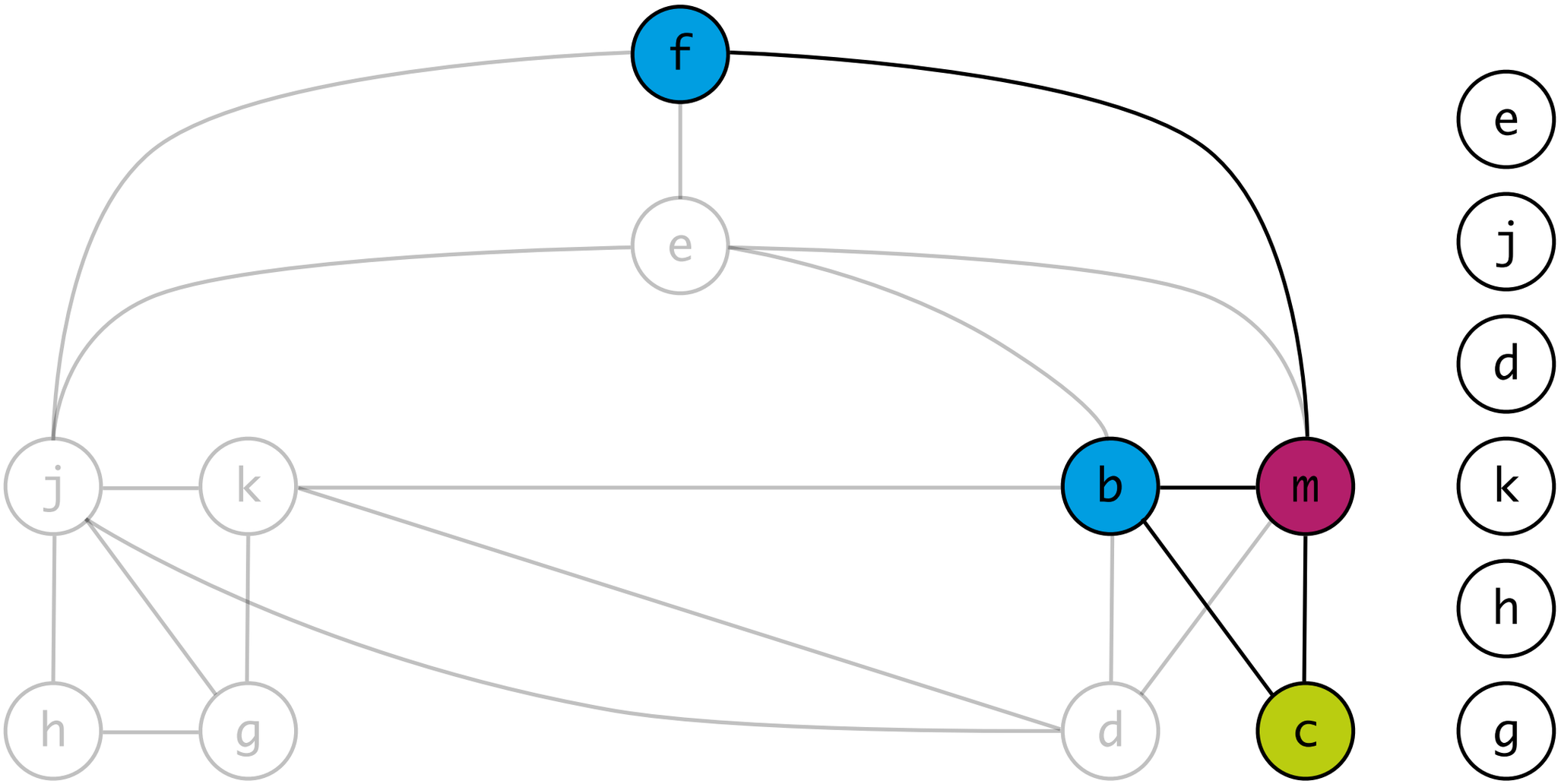
Graph colouring

example with 4 colours



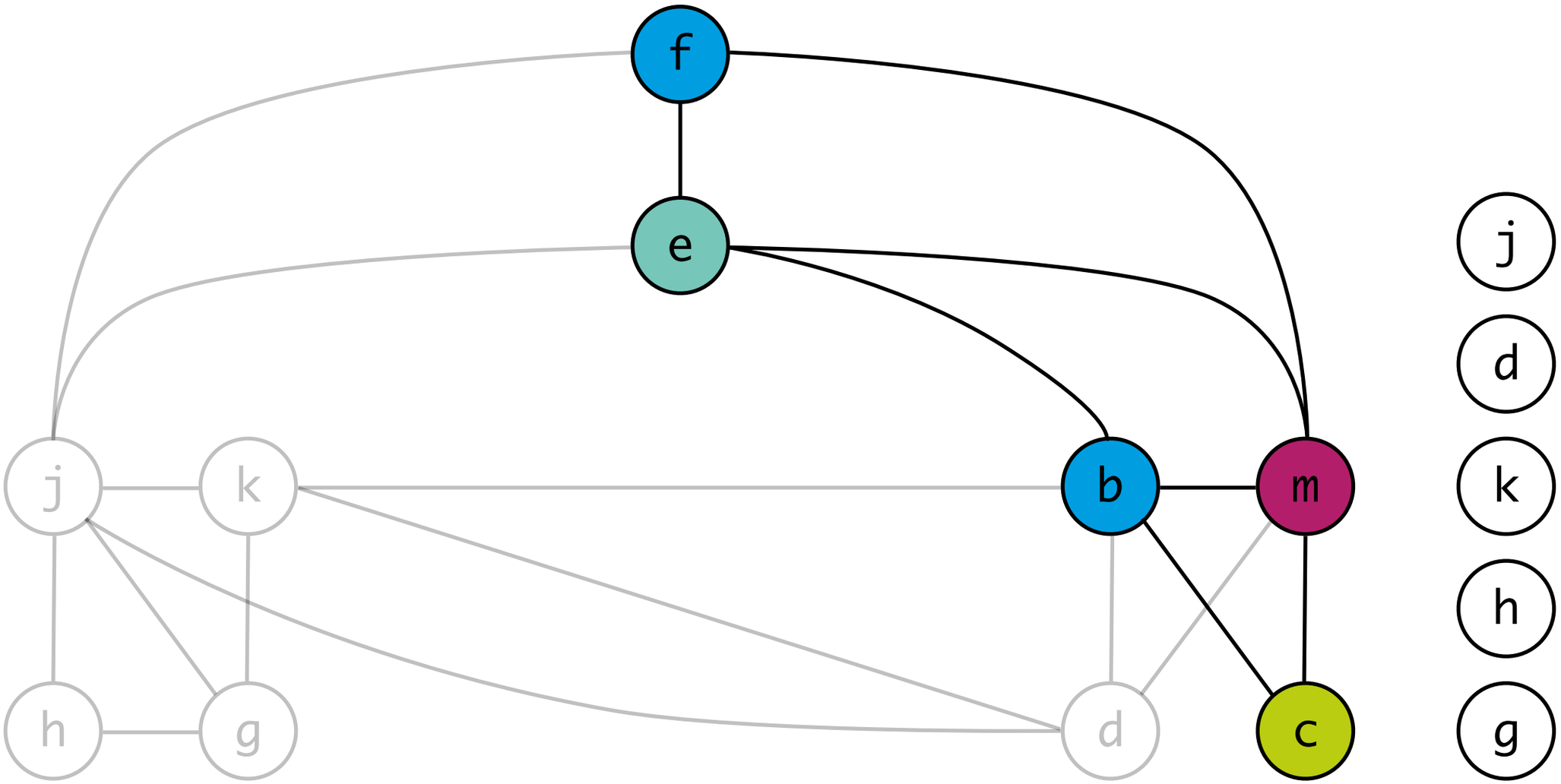
Graph colouring

example with 4 colours



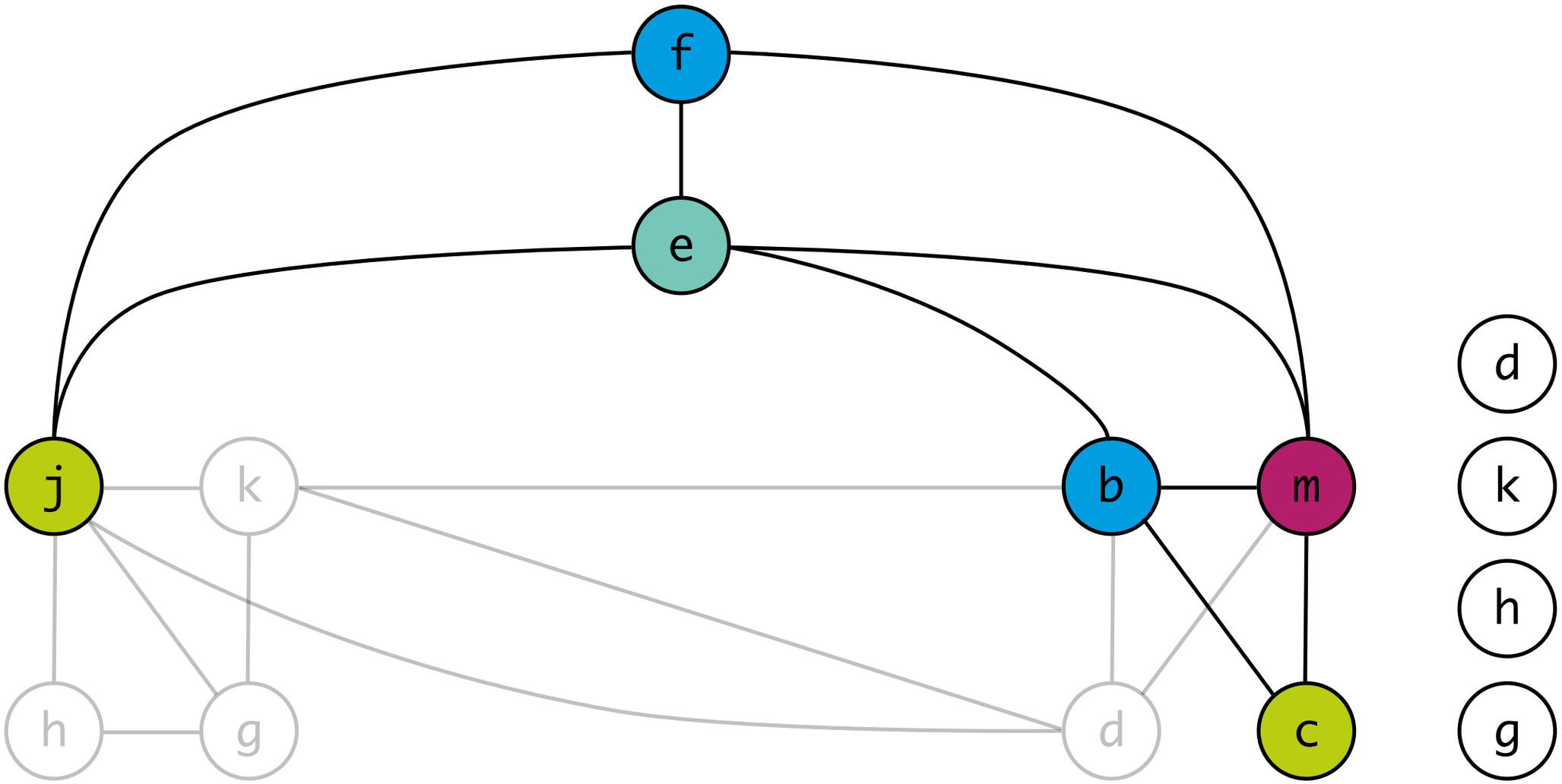
Graph colouring

example with 4 colours



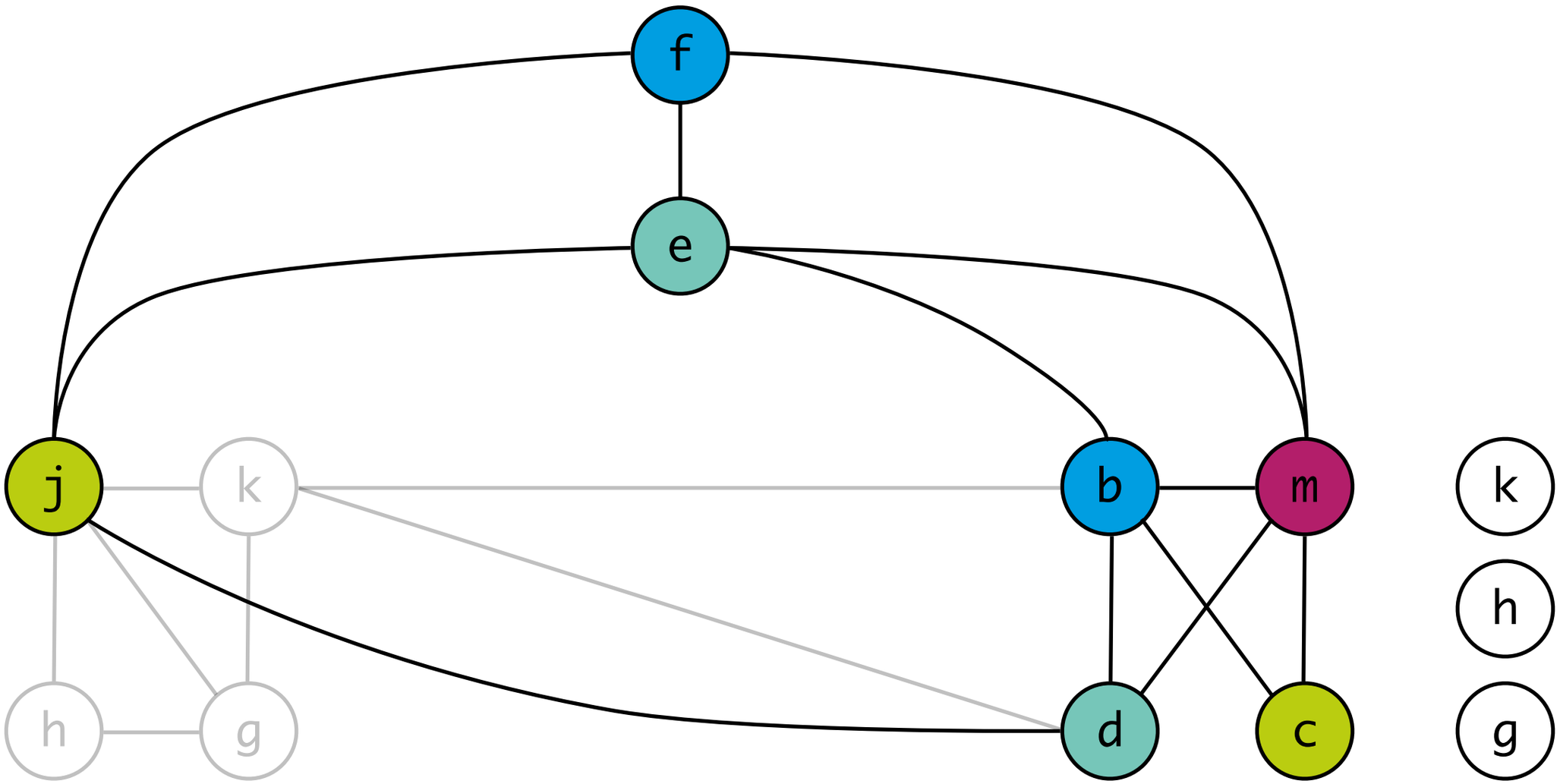
Graph colouring

example with 4 colours



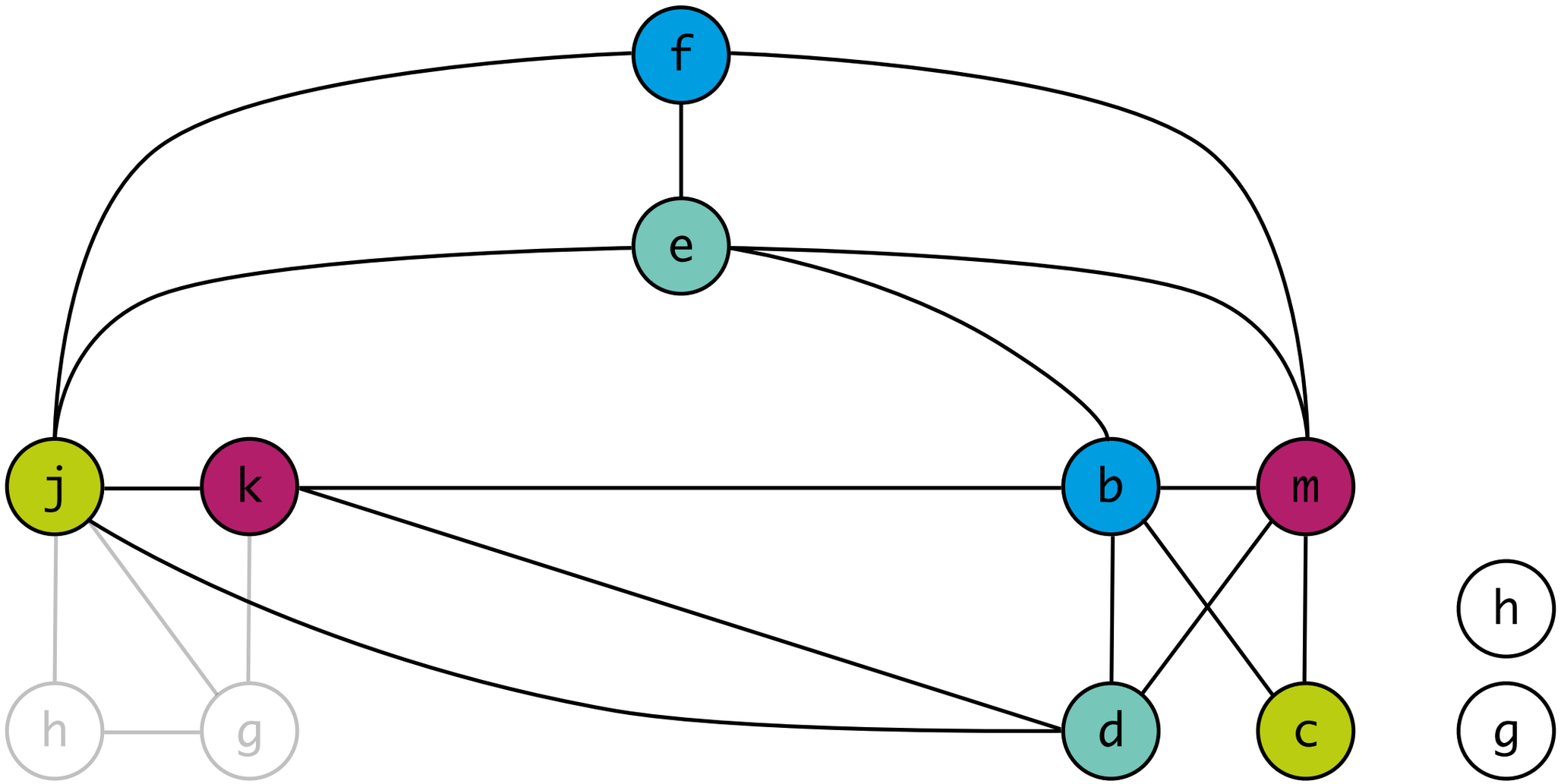
Graph colouring

example with 4 colours



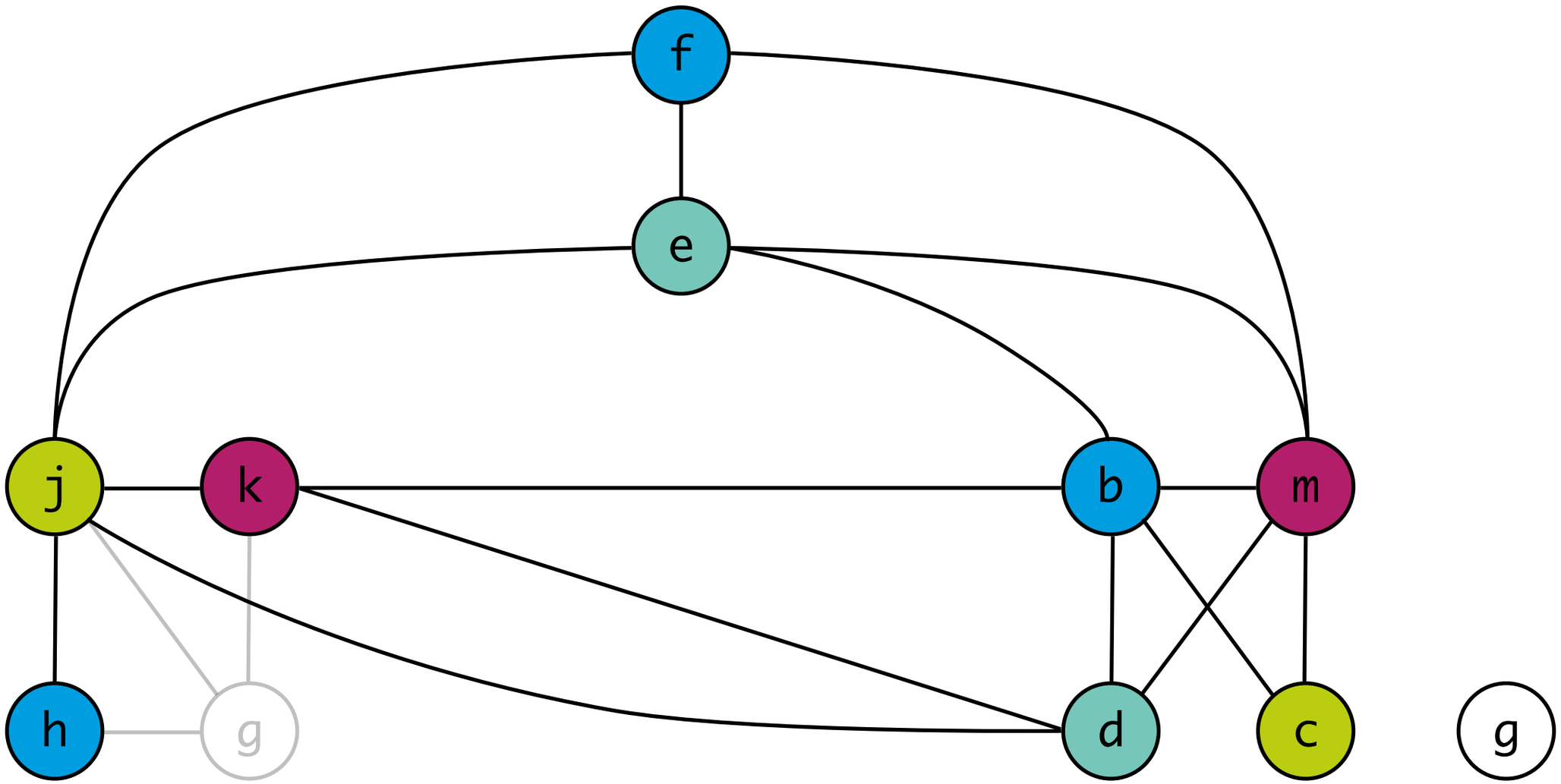
Graph colouring

example with 4 colours



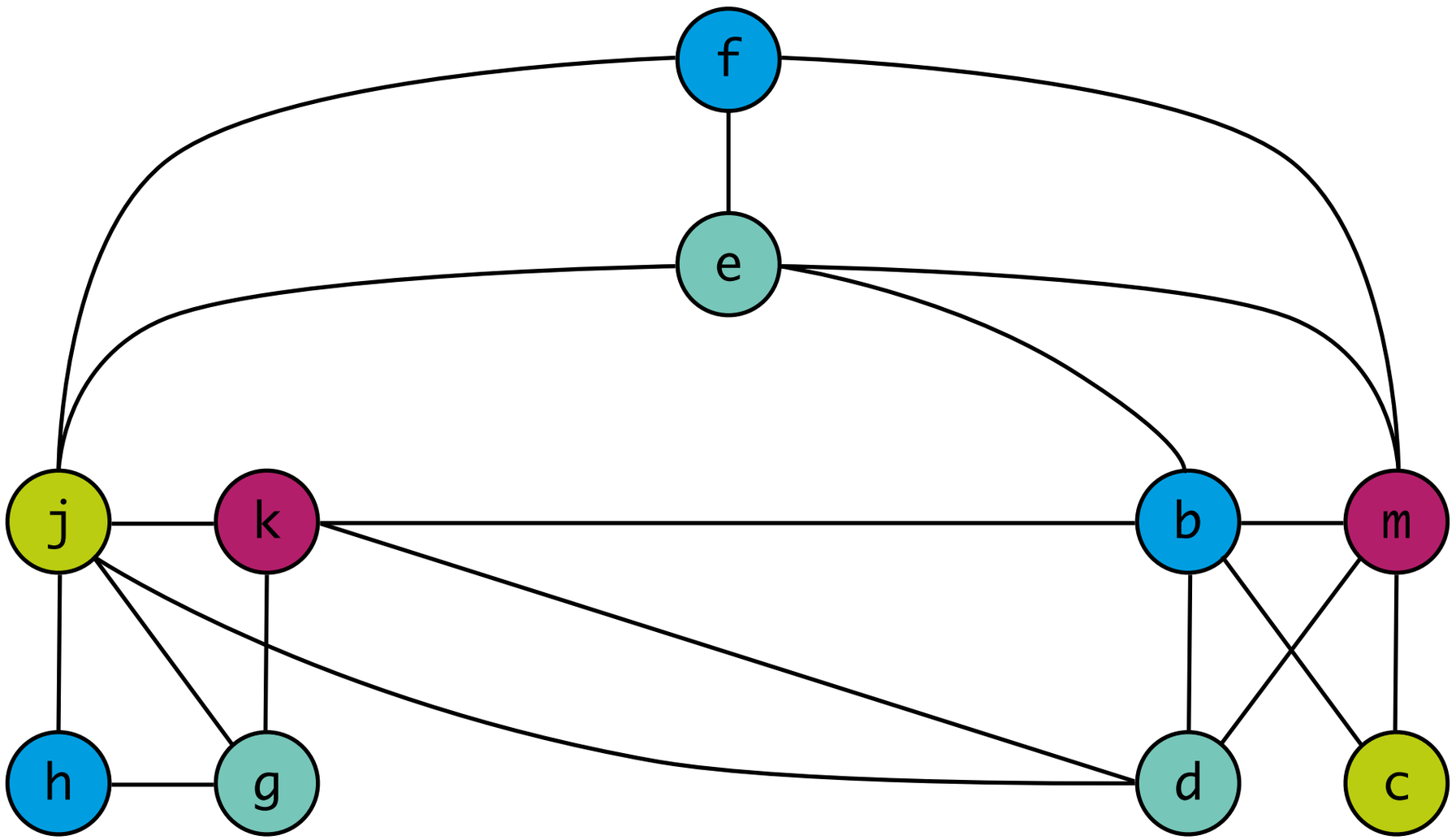
Graph colouring

example with 4 colours



Graph colouring

example with 4 colours



Optimistic colouring

steps

simplify

remove node of insignificant degree (fewer than k edges)

spill

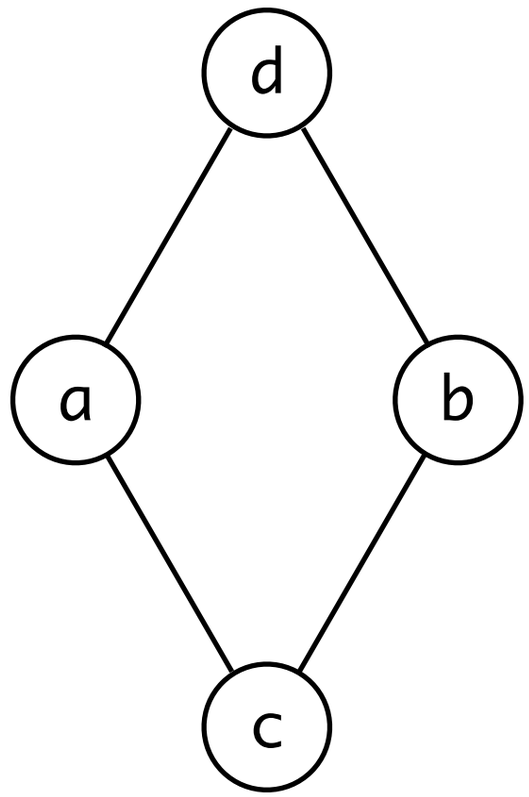
remove node of significant degree (k or more edges)

select

add node, select colour

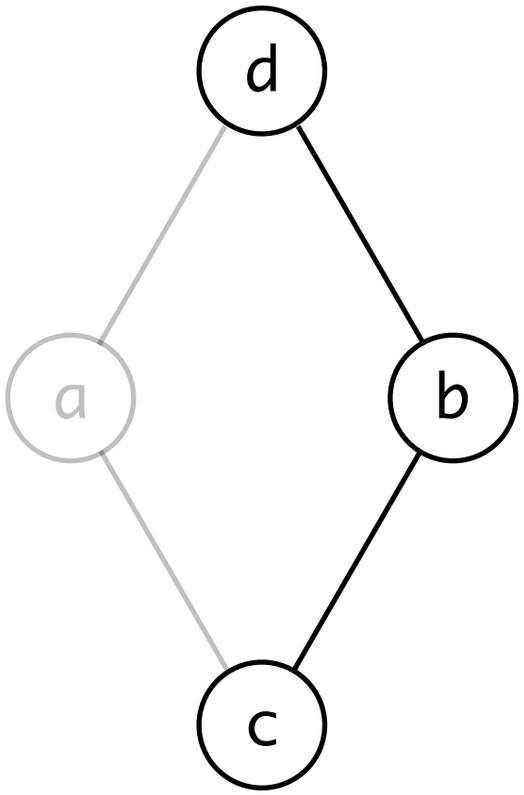
Optimistic colouring

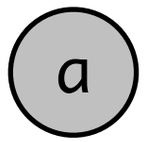
example with 2 colours



Optimistic colouring

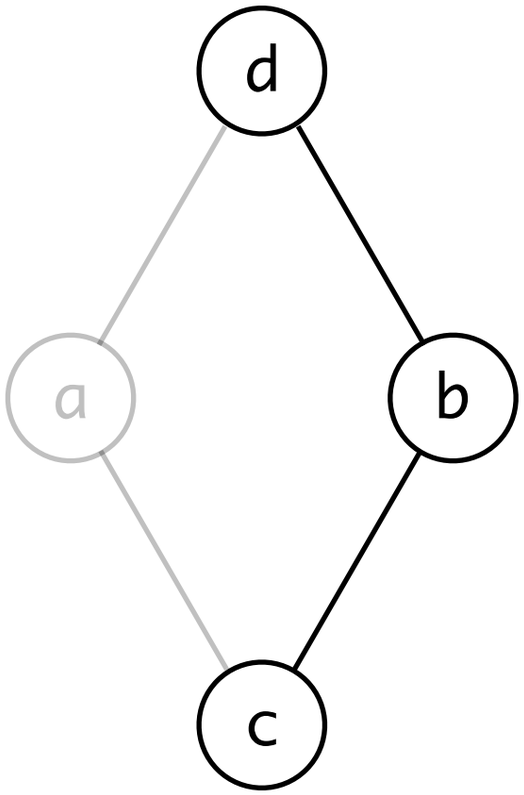
example with 2 colours



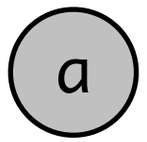


Optimistic colouring

example with 2 colours

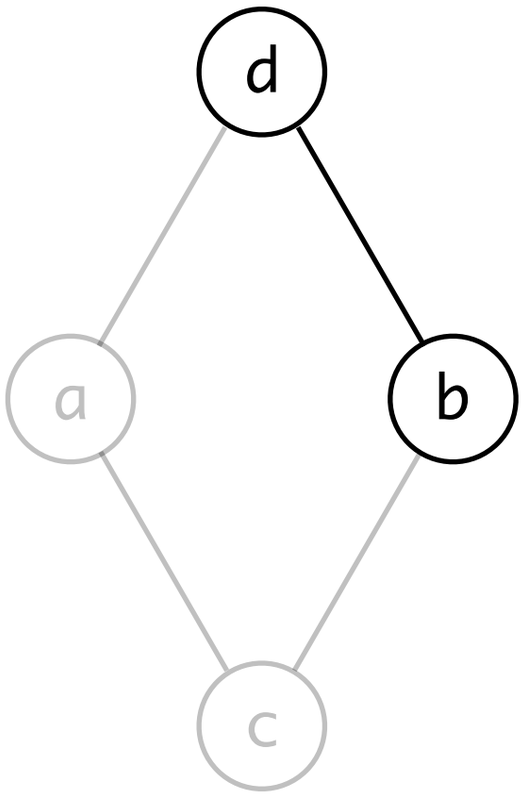


potential spill

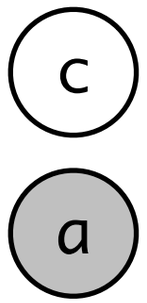


Optimistic colouring

example with 2 colours

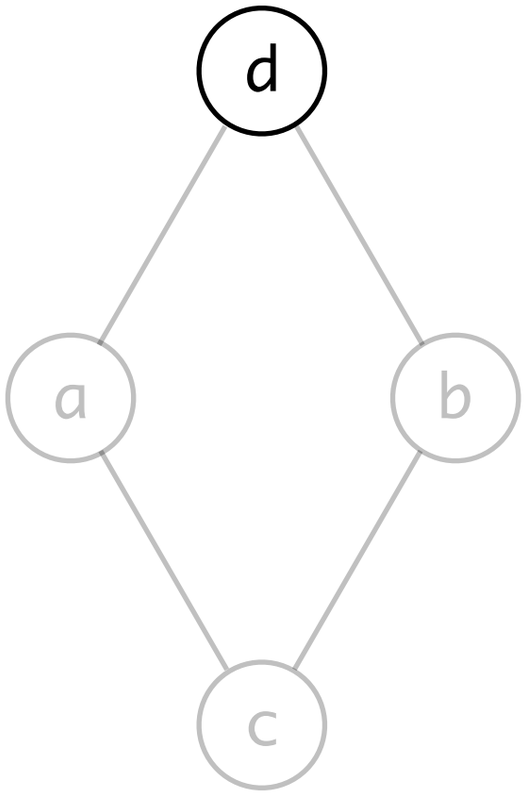


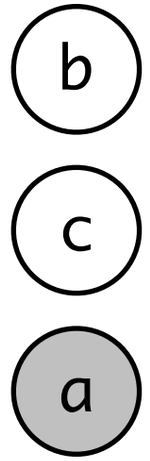
potential spill



Optimistic colouring

example with 2 colours

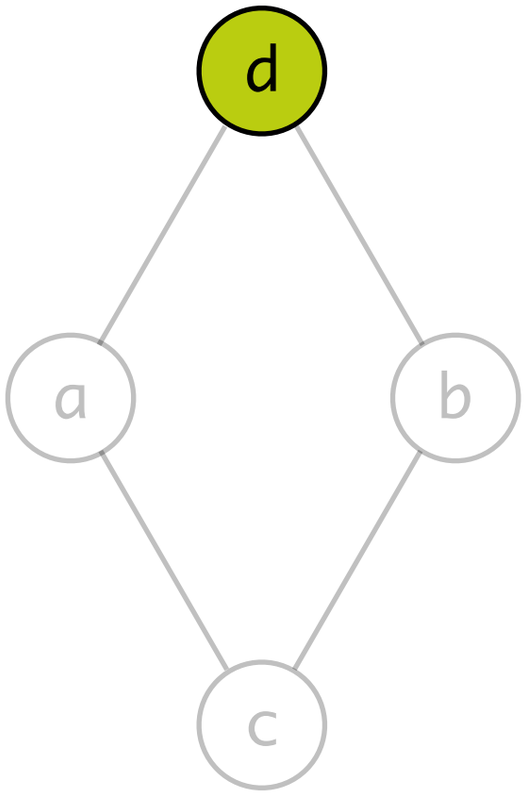




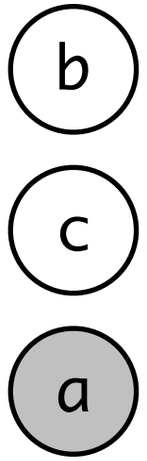
potential spill

Optimistic colouring

example with 2 colours

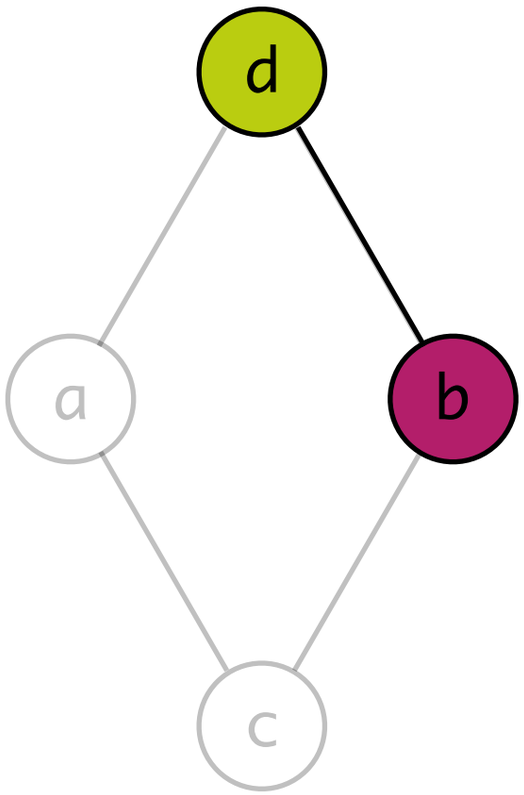


potential spill

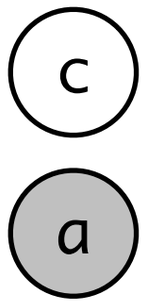


Optimistic colouring

example with 2 colours

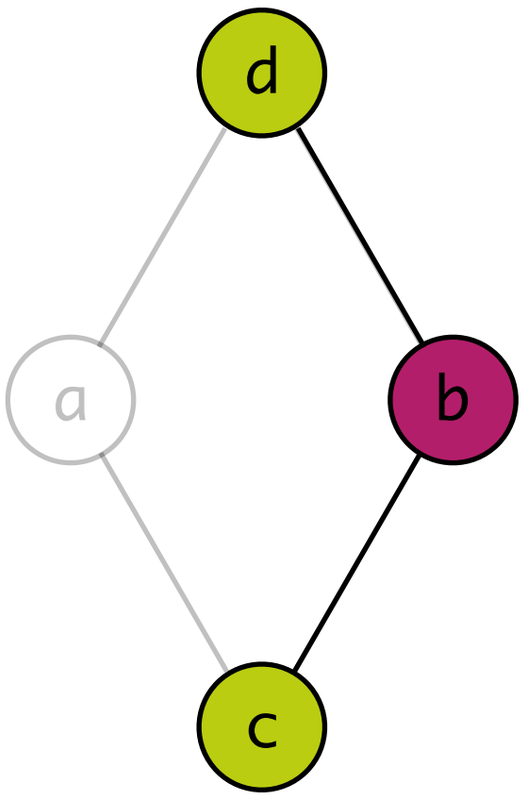


potential spill

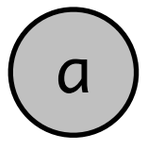


Optimistic colouring

example with 2 colours

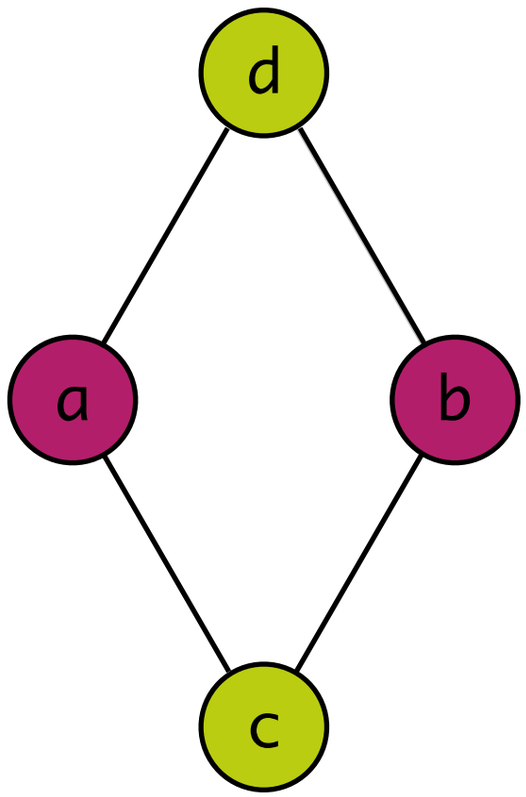


potential spill



Optimistic colouring

example with 2 colours



Spilling
steps

simplify

remove node of insignificant degree (less than k edges)

spill

remove node of significant degree (k or more edges)

select

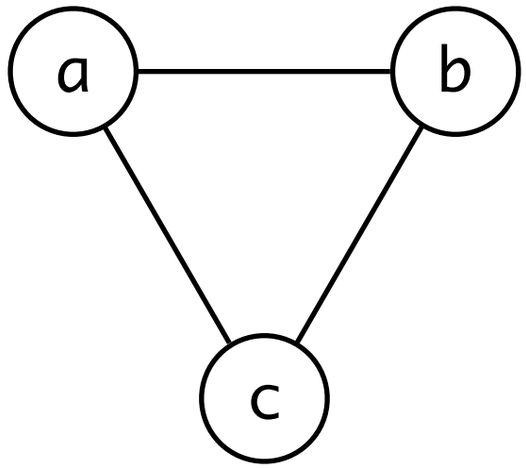
add node, select colour

actual spill

start over

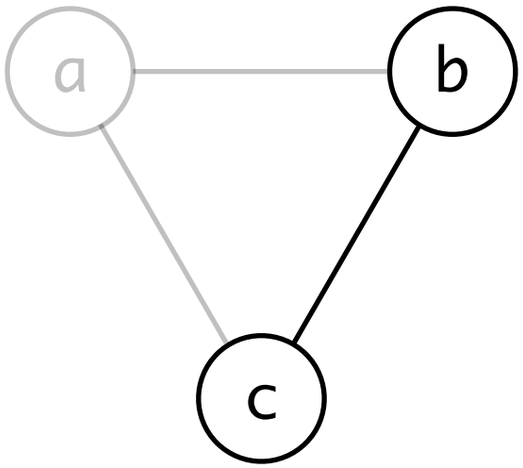
Spilling

example with 2 colours

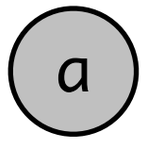


Spilling

example with 2 colours

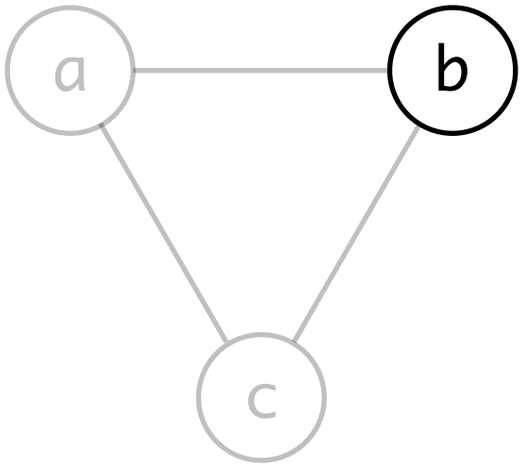


potential spill

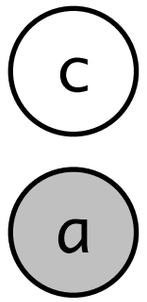


Spilling

example with 2 colours

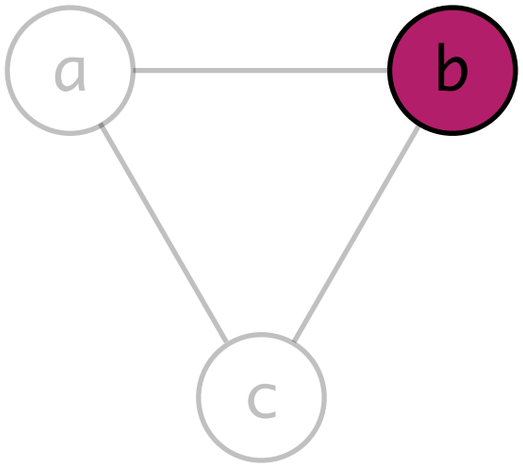


potential spill

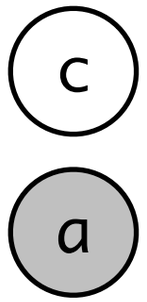


Spilling

example with 2 colours

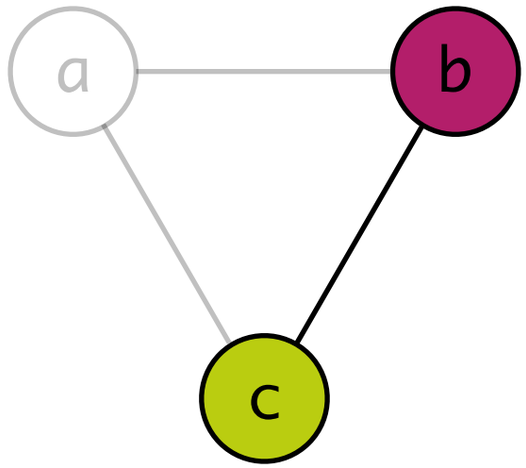


potential spill

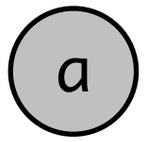


Spilling

example with 2 colours

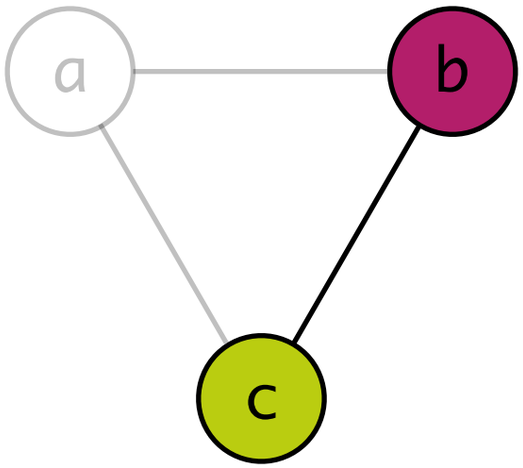


potential spill

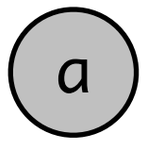


Spilling

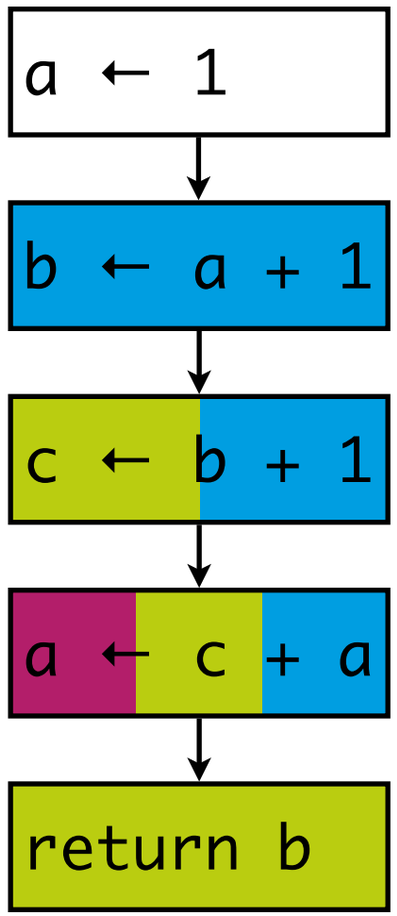
example with 2 colours



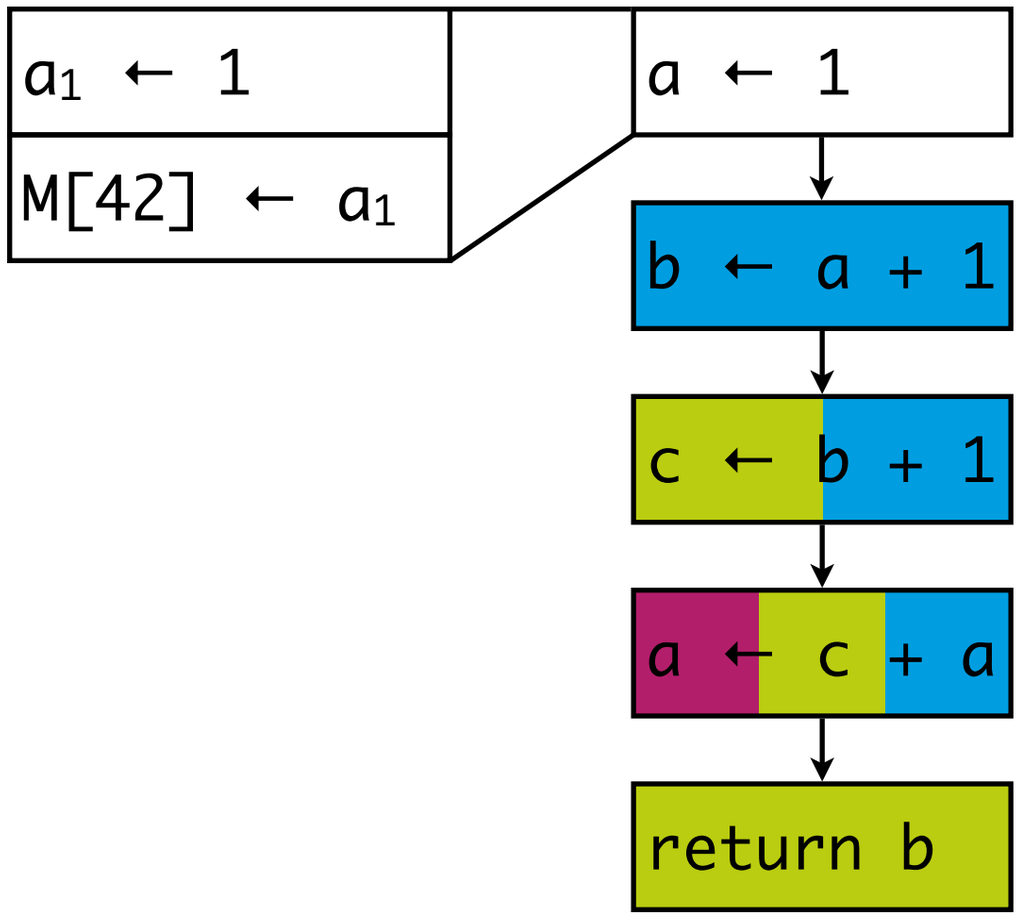
actual spill



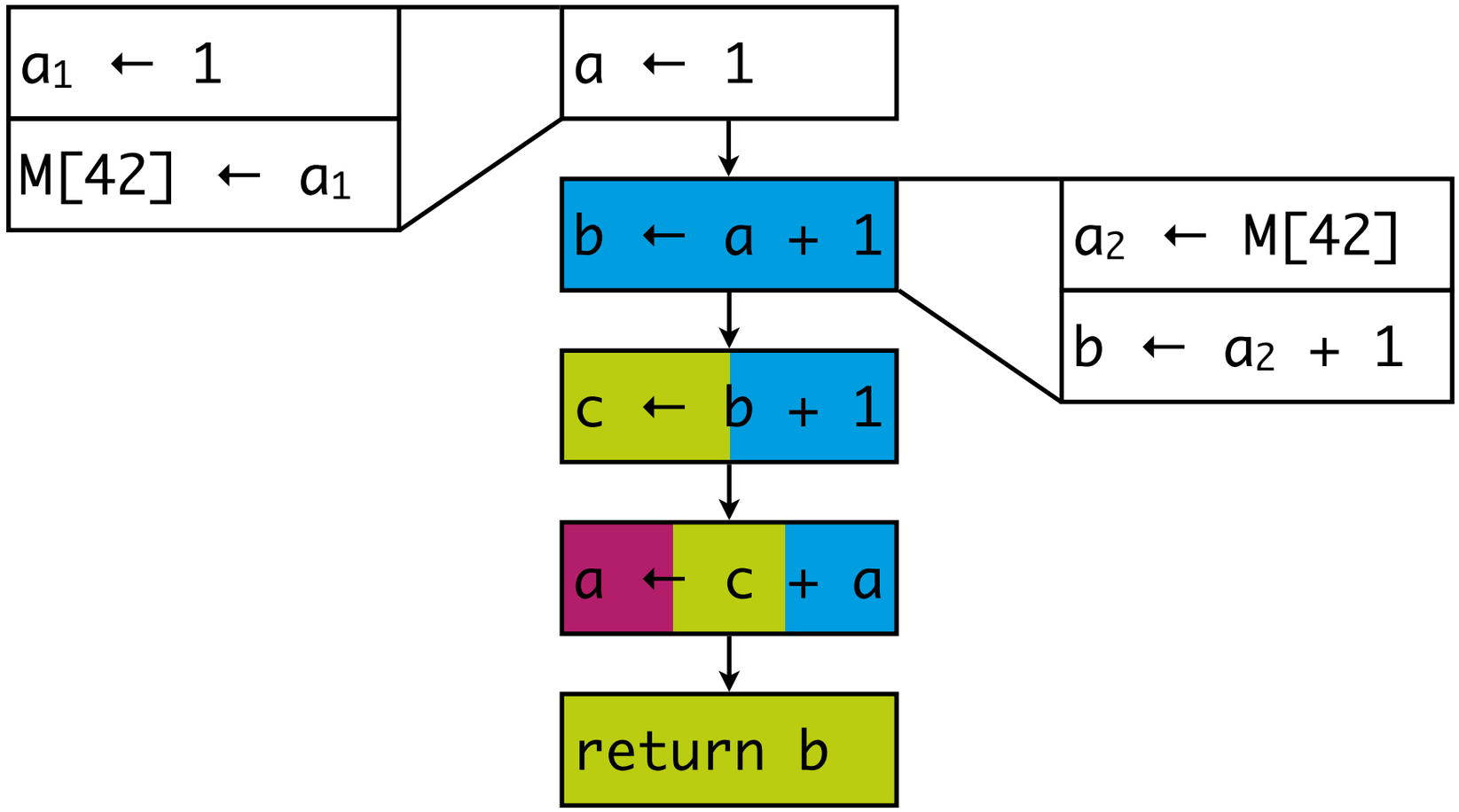
Spilling
example



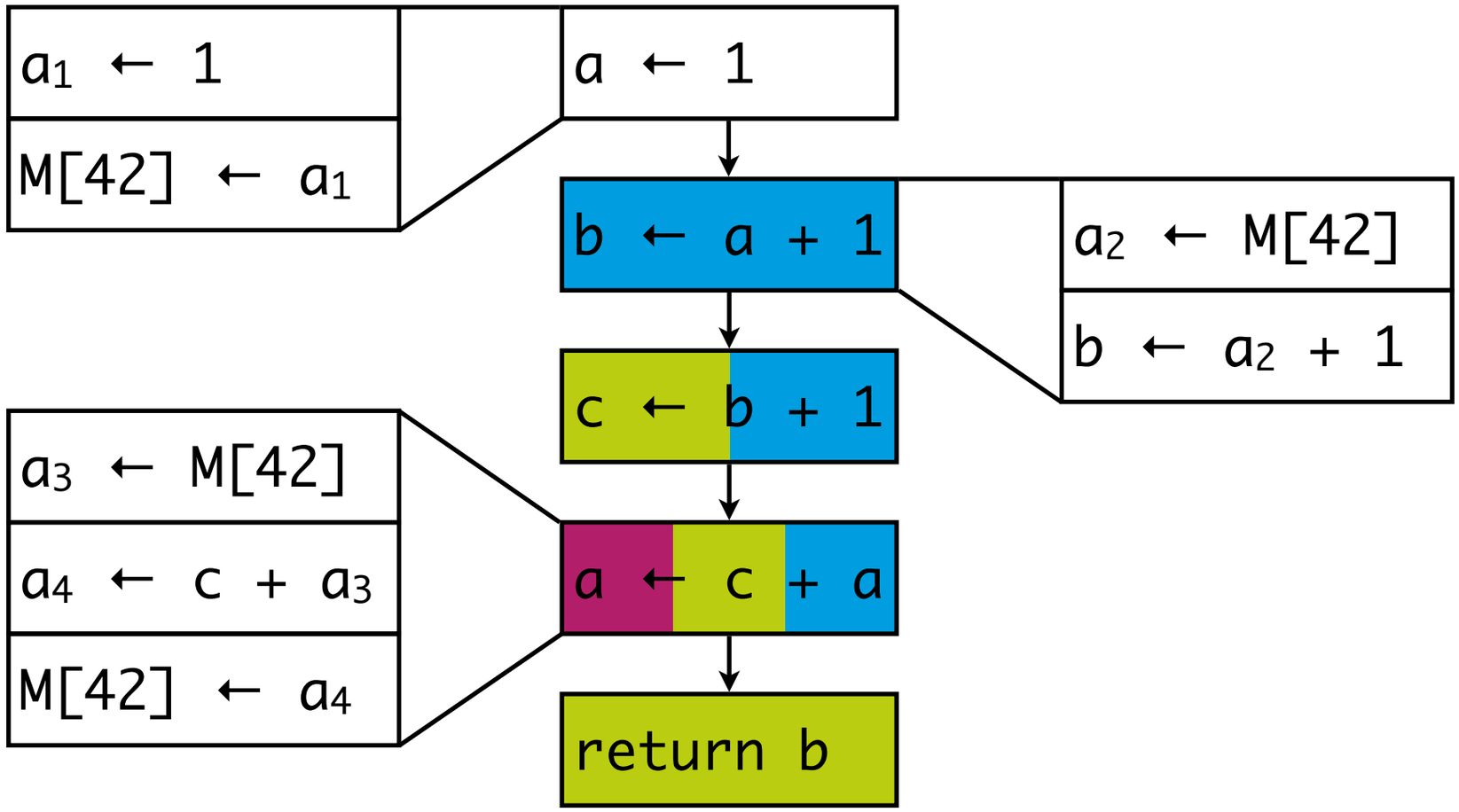
Spilling
example



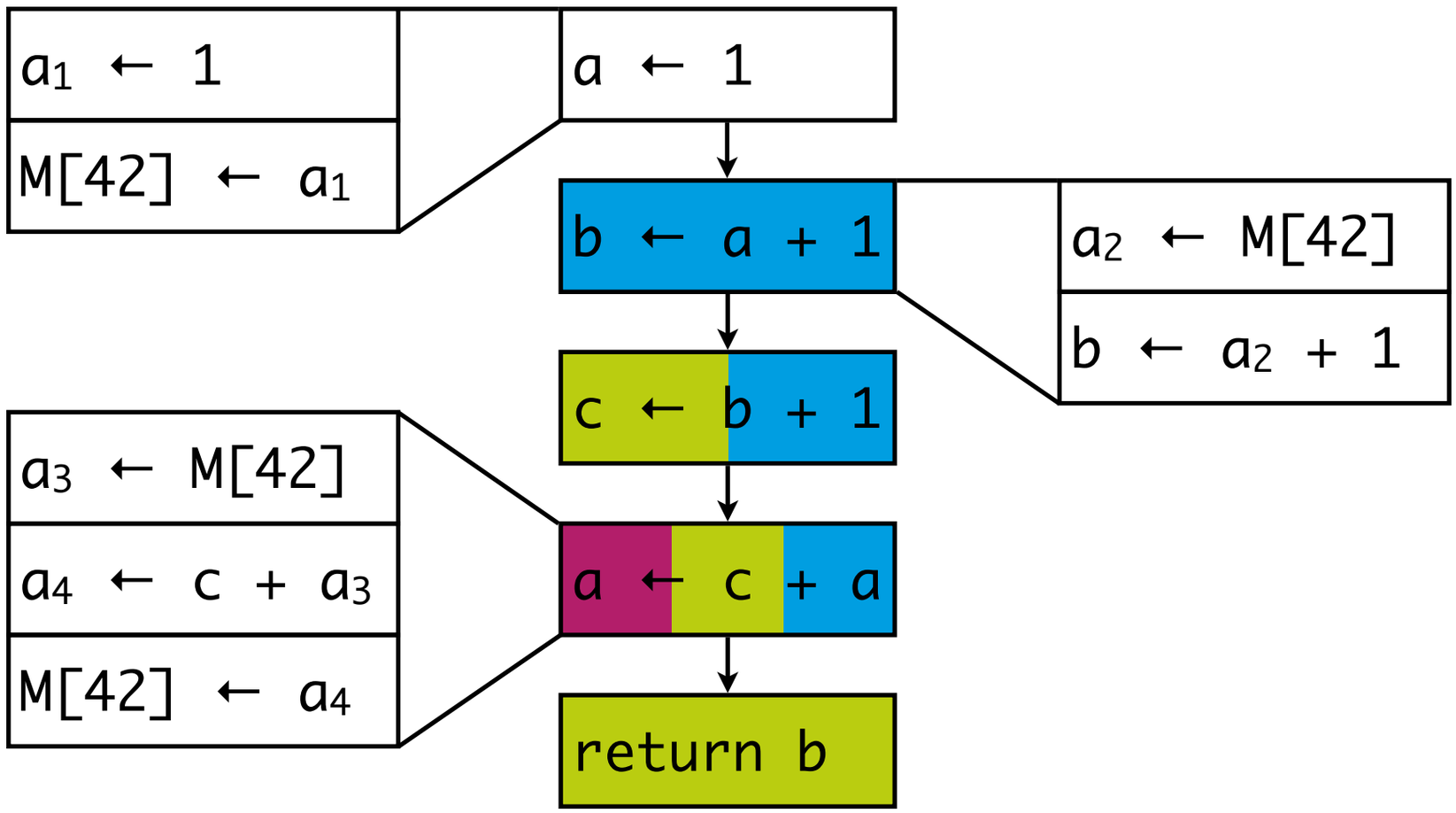
Spilling
example



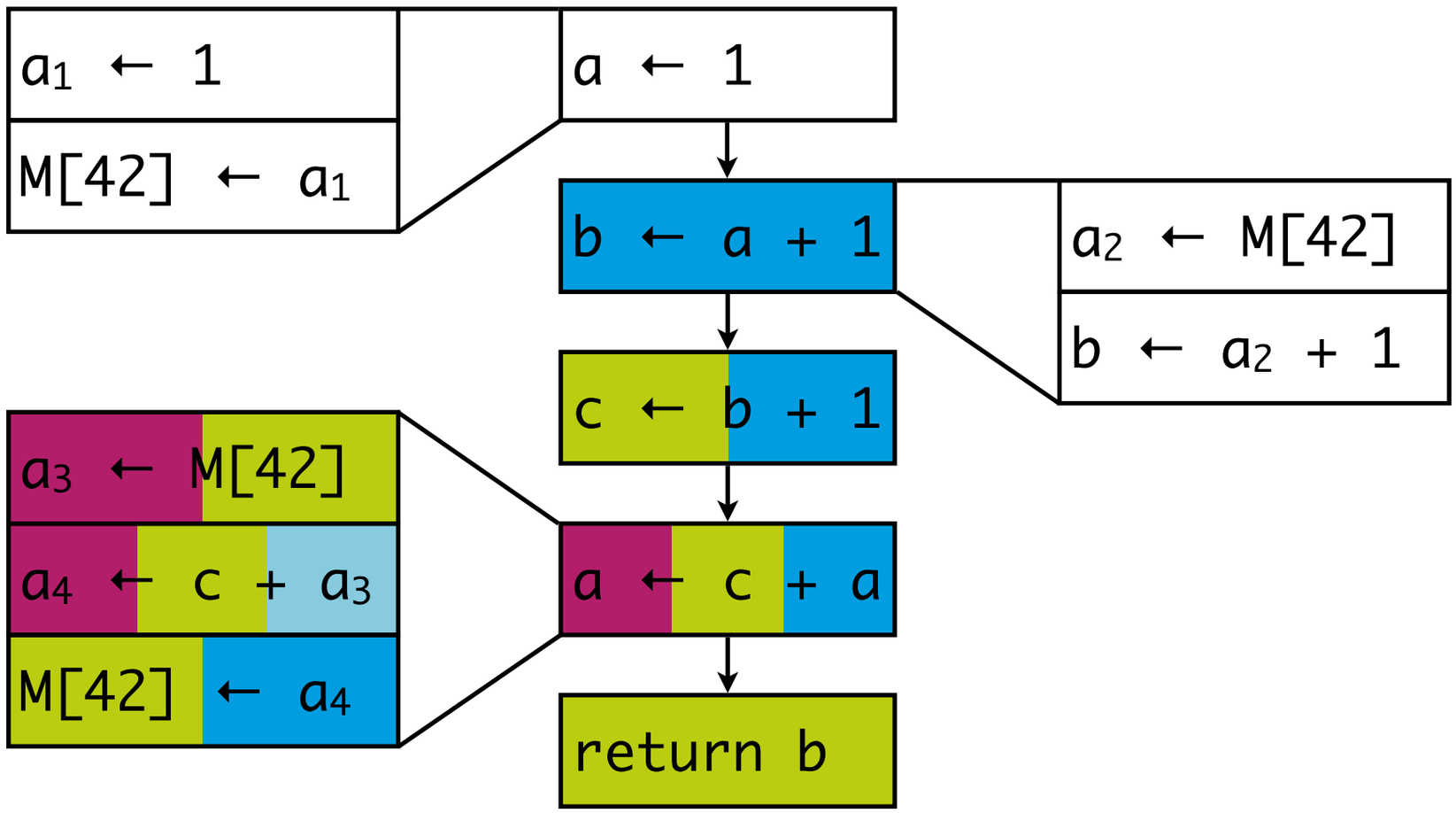
Spilling
example



Spilling
example

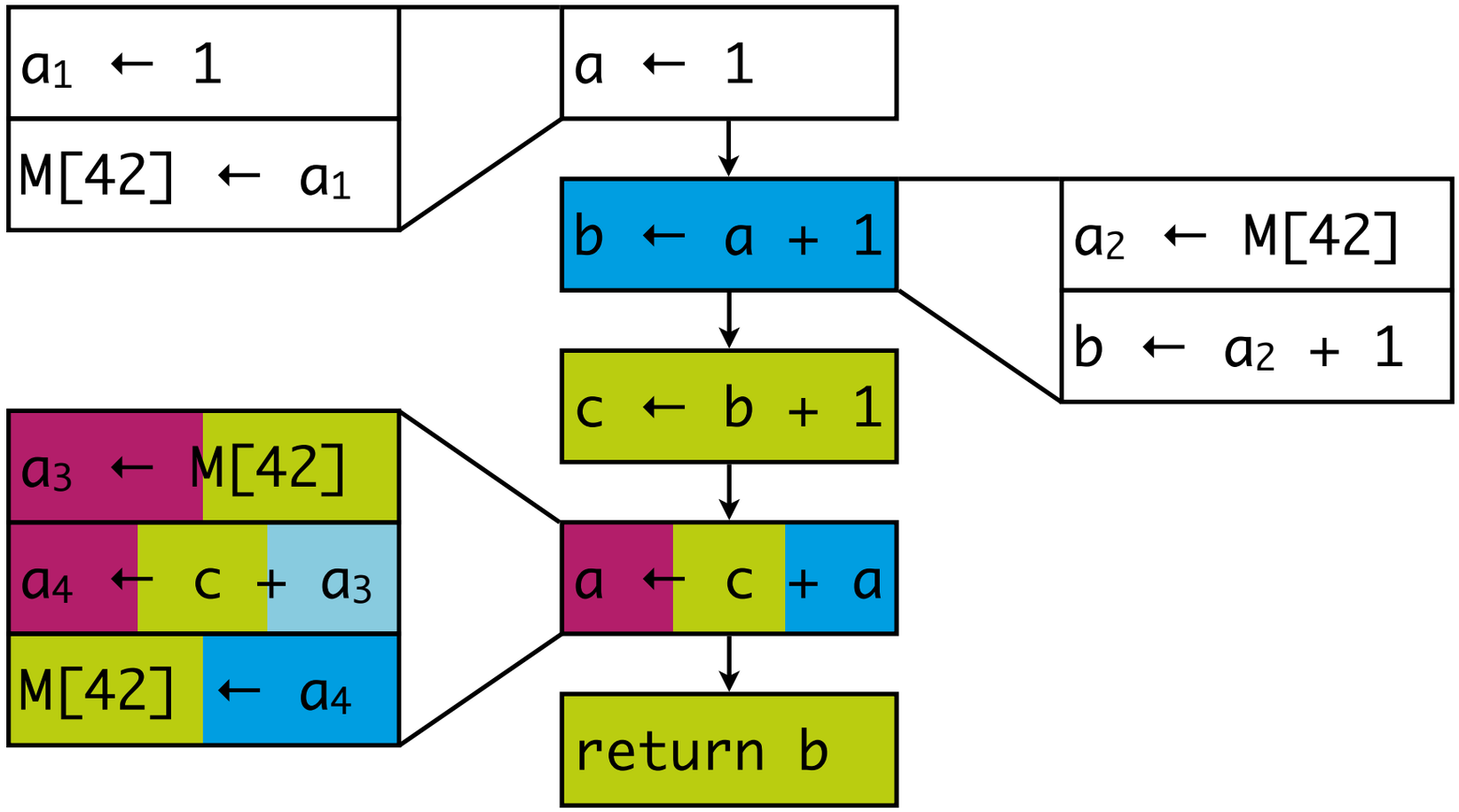


Spilling
example

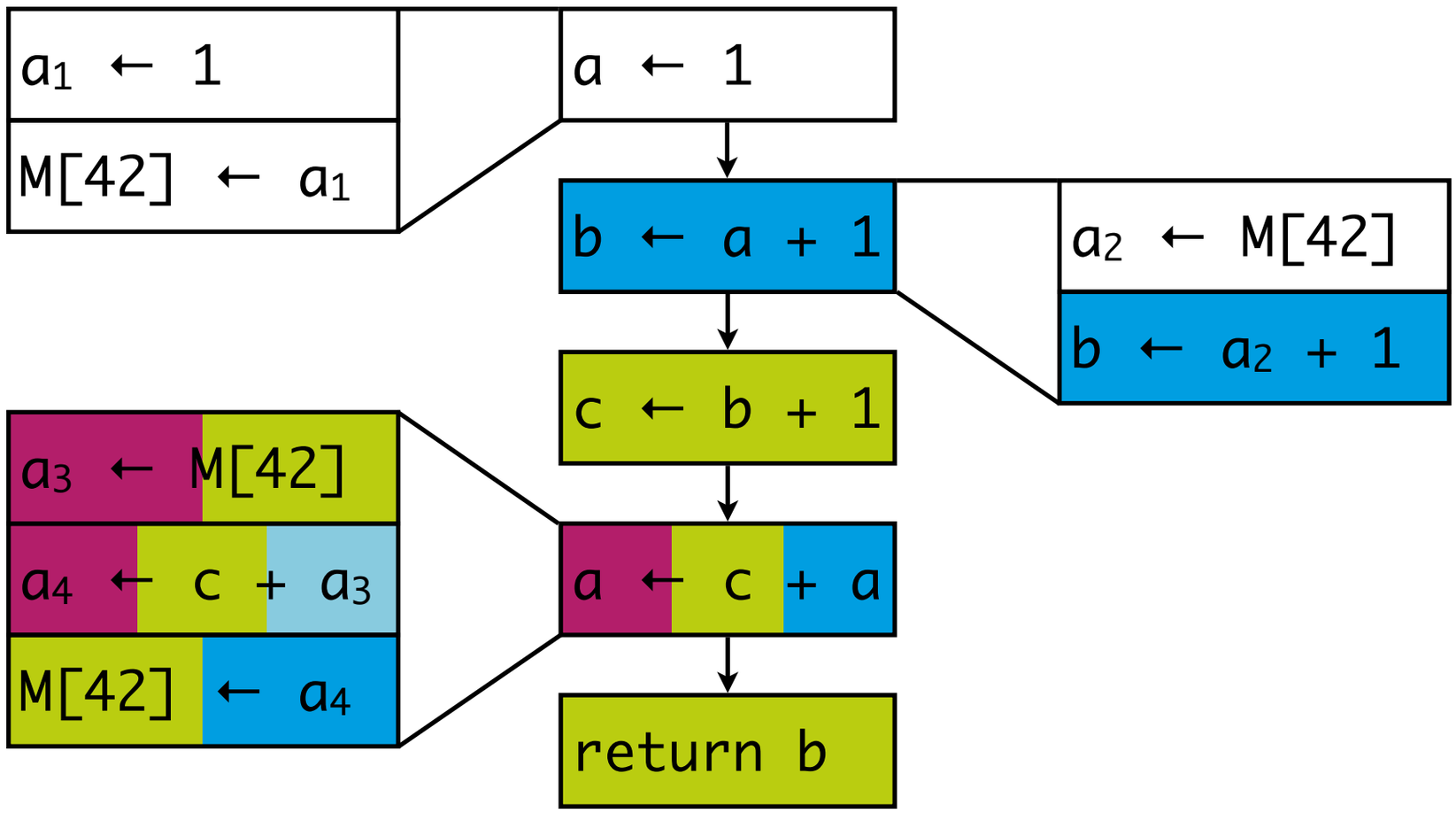


Spilling

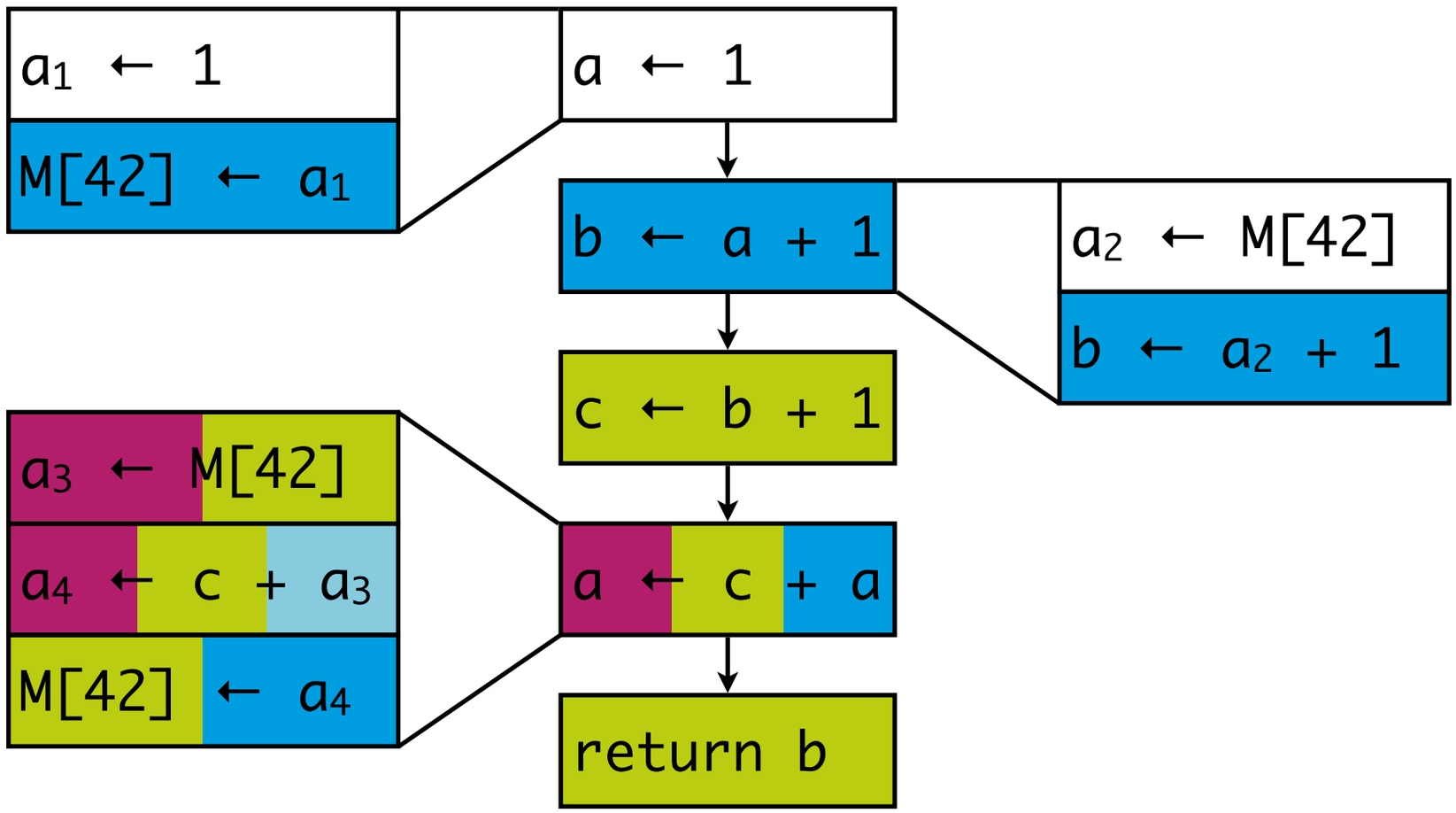
example



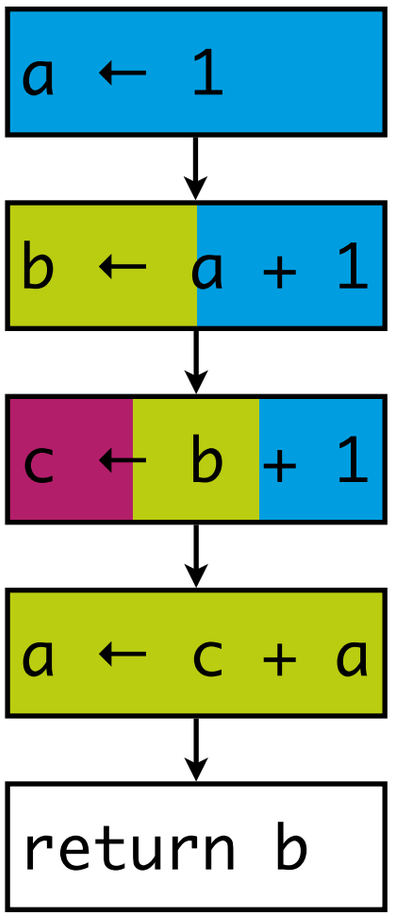
Spilling
example



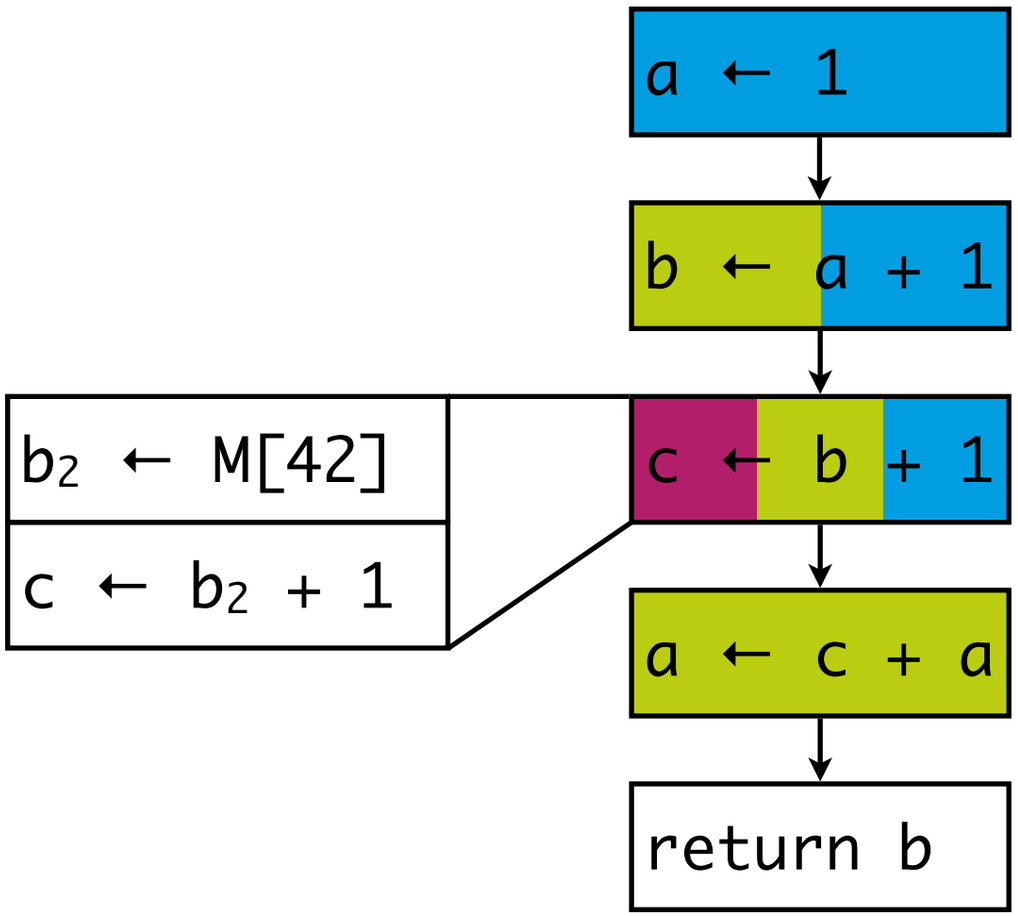
Spilling
example



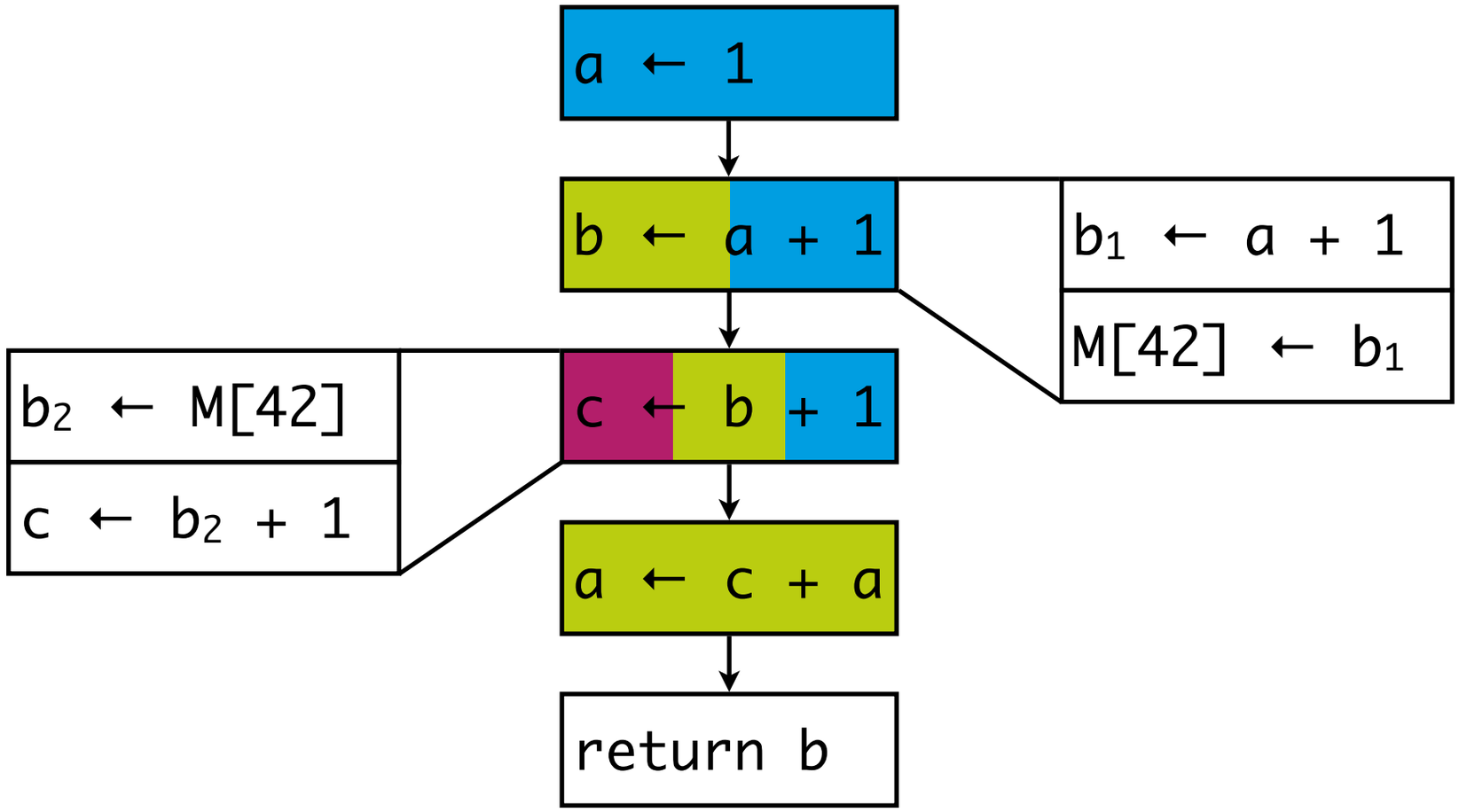
Spilling
example



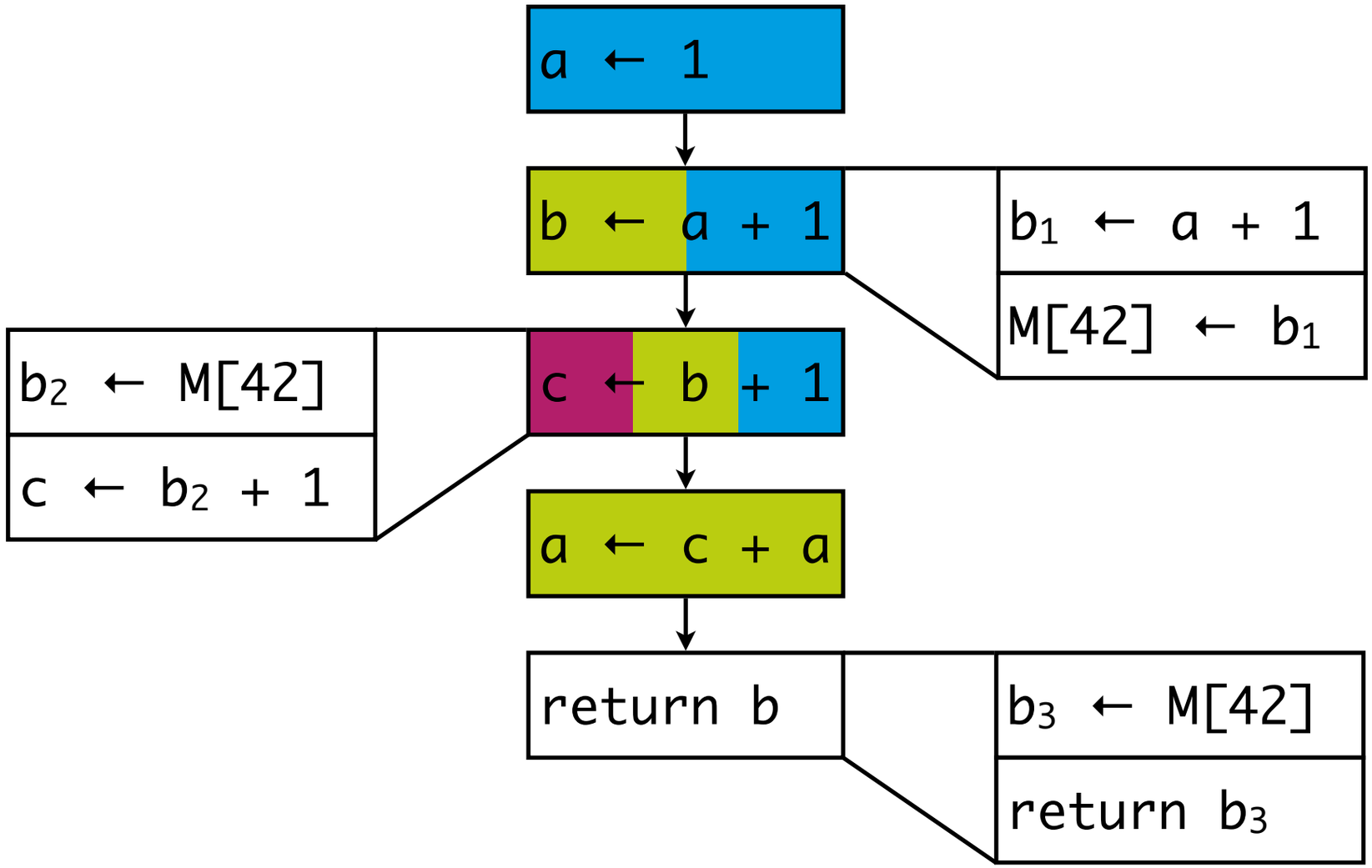
Spilling
example



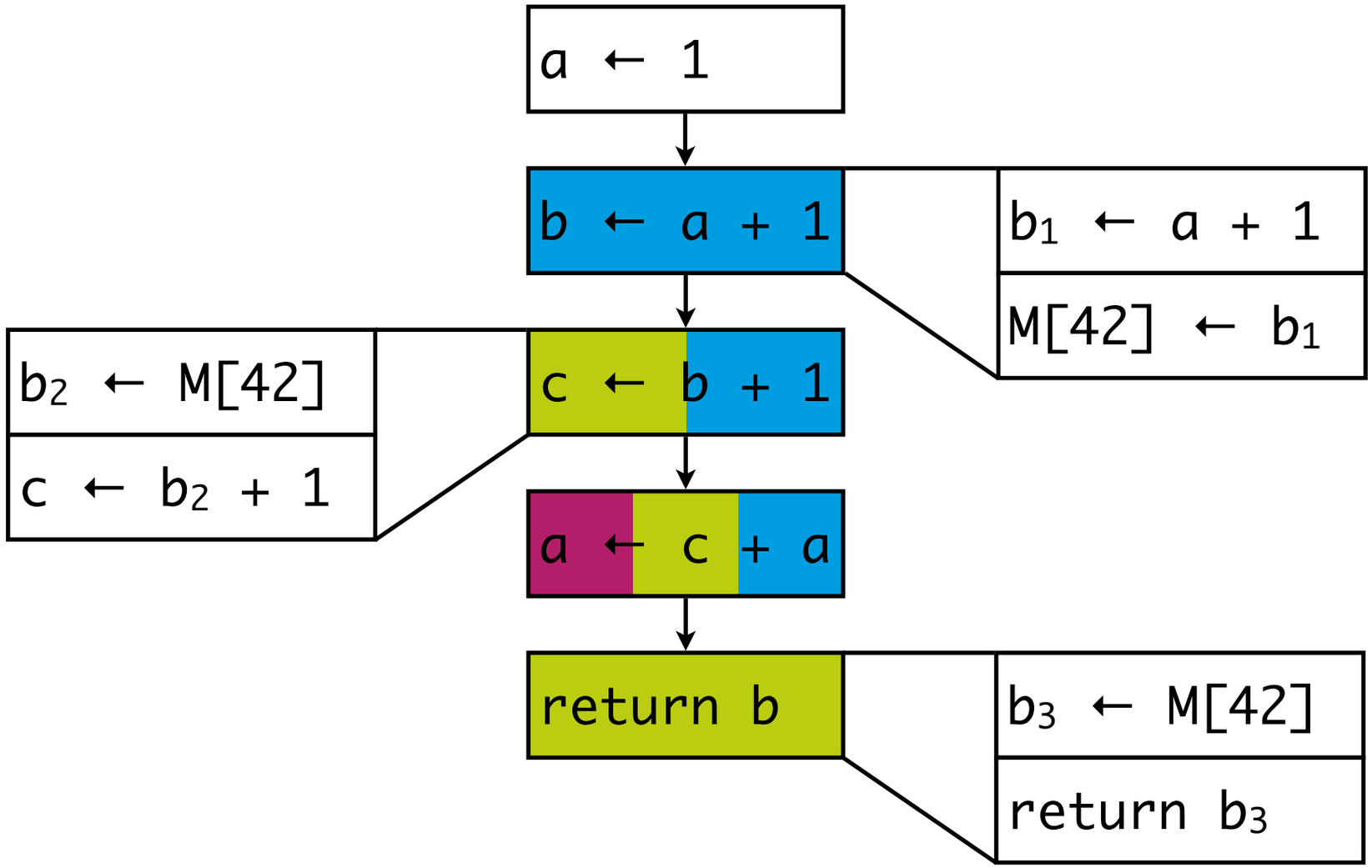
Spilling
example



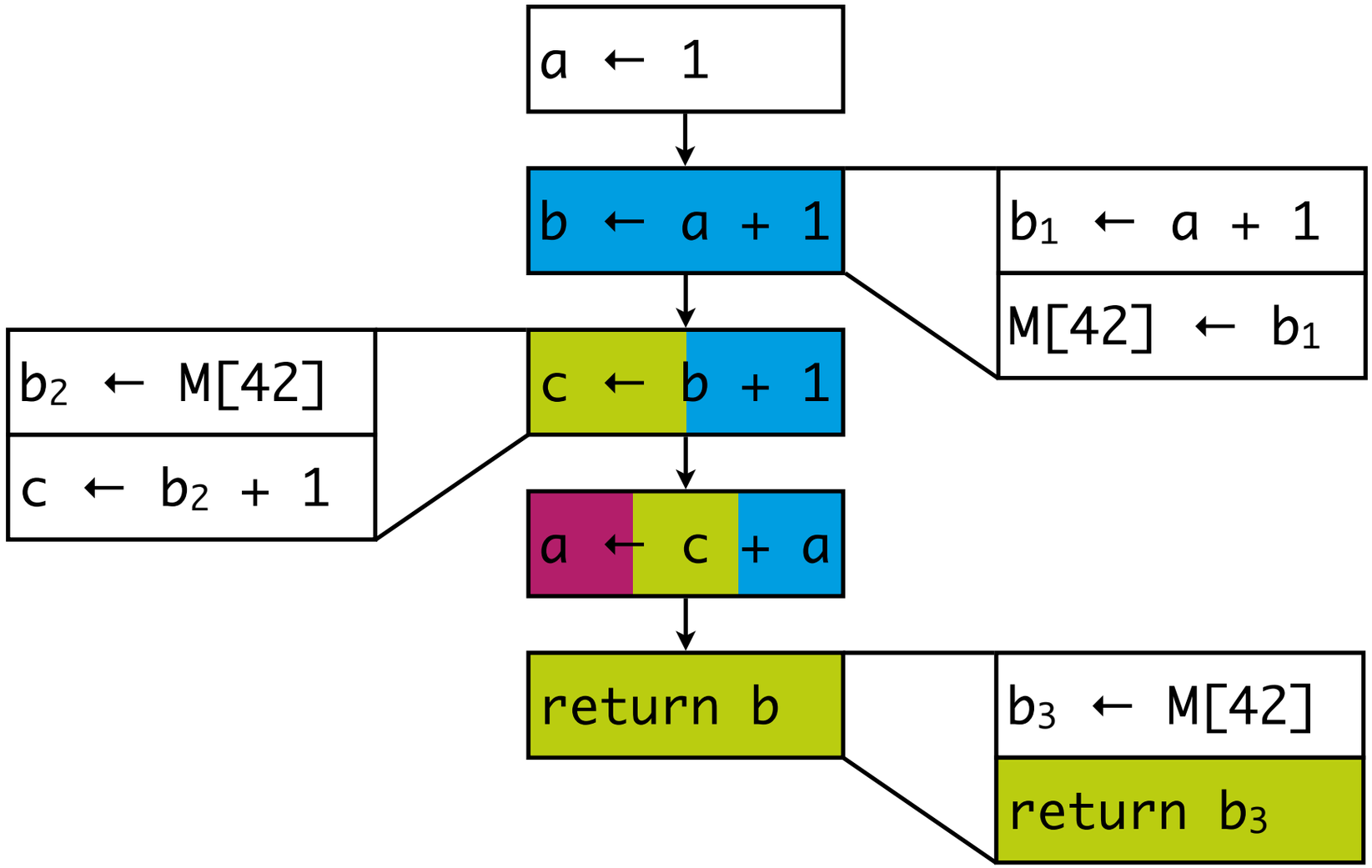
Spilling
example



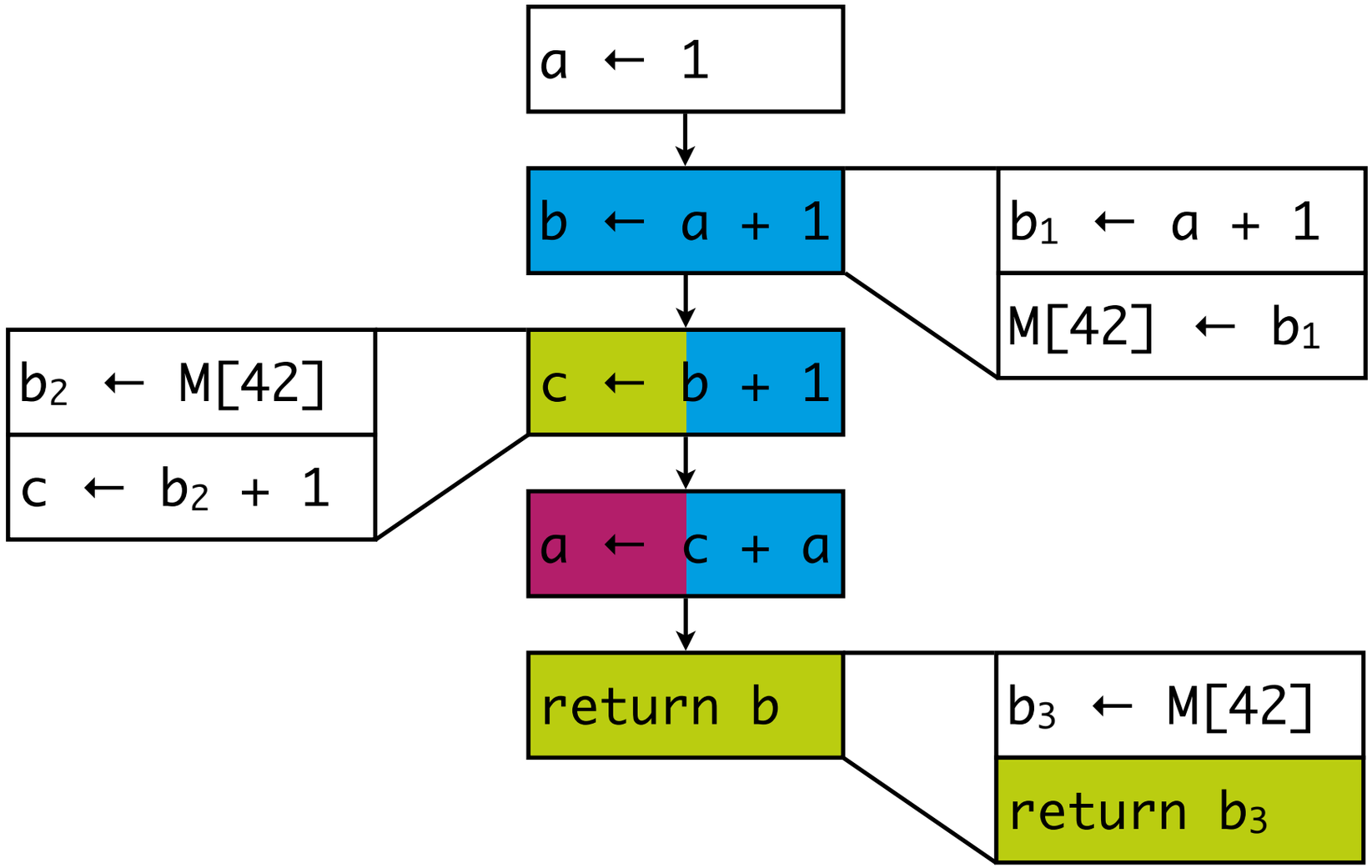
Spilling
example



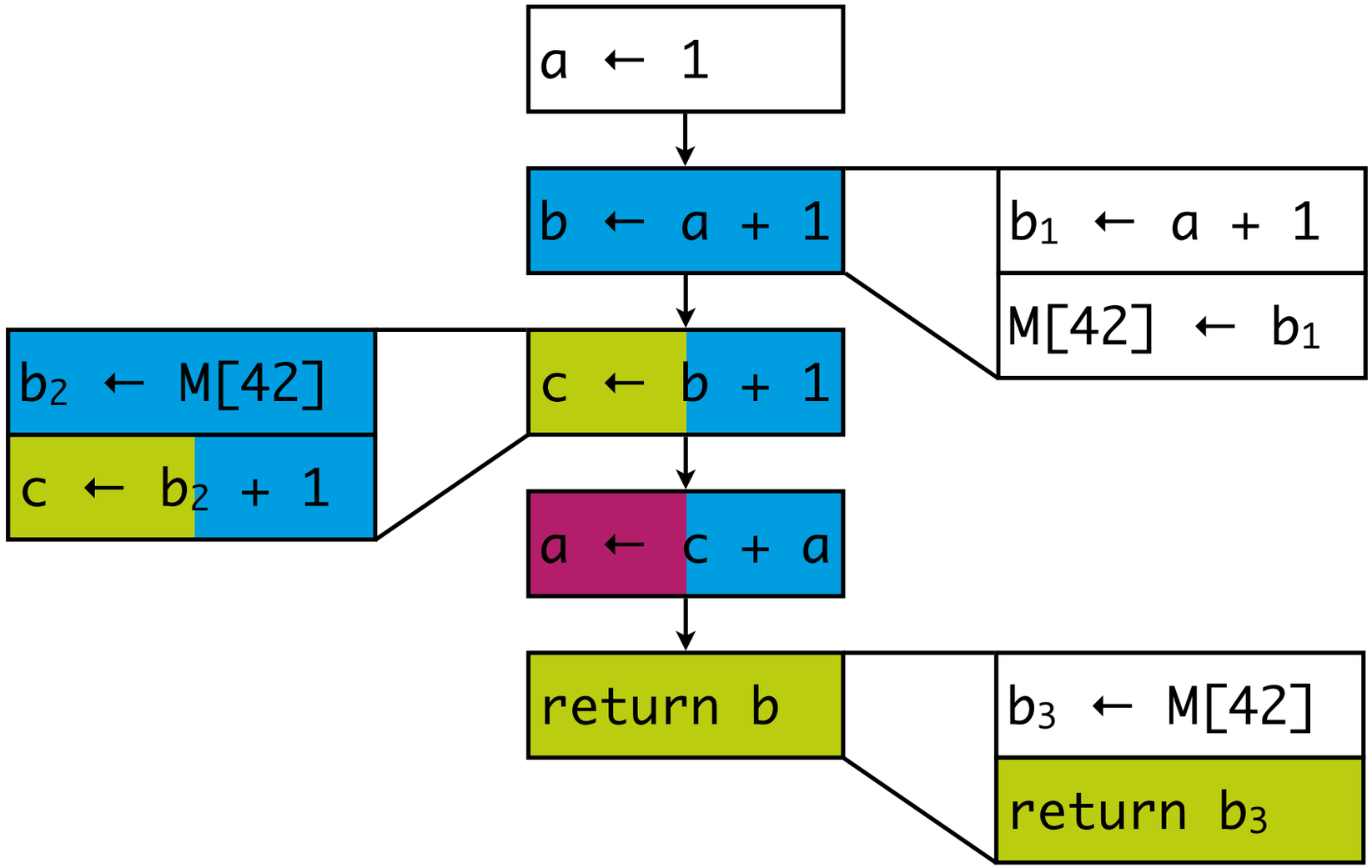
Spilling
example



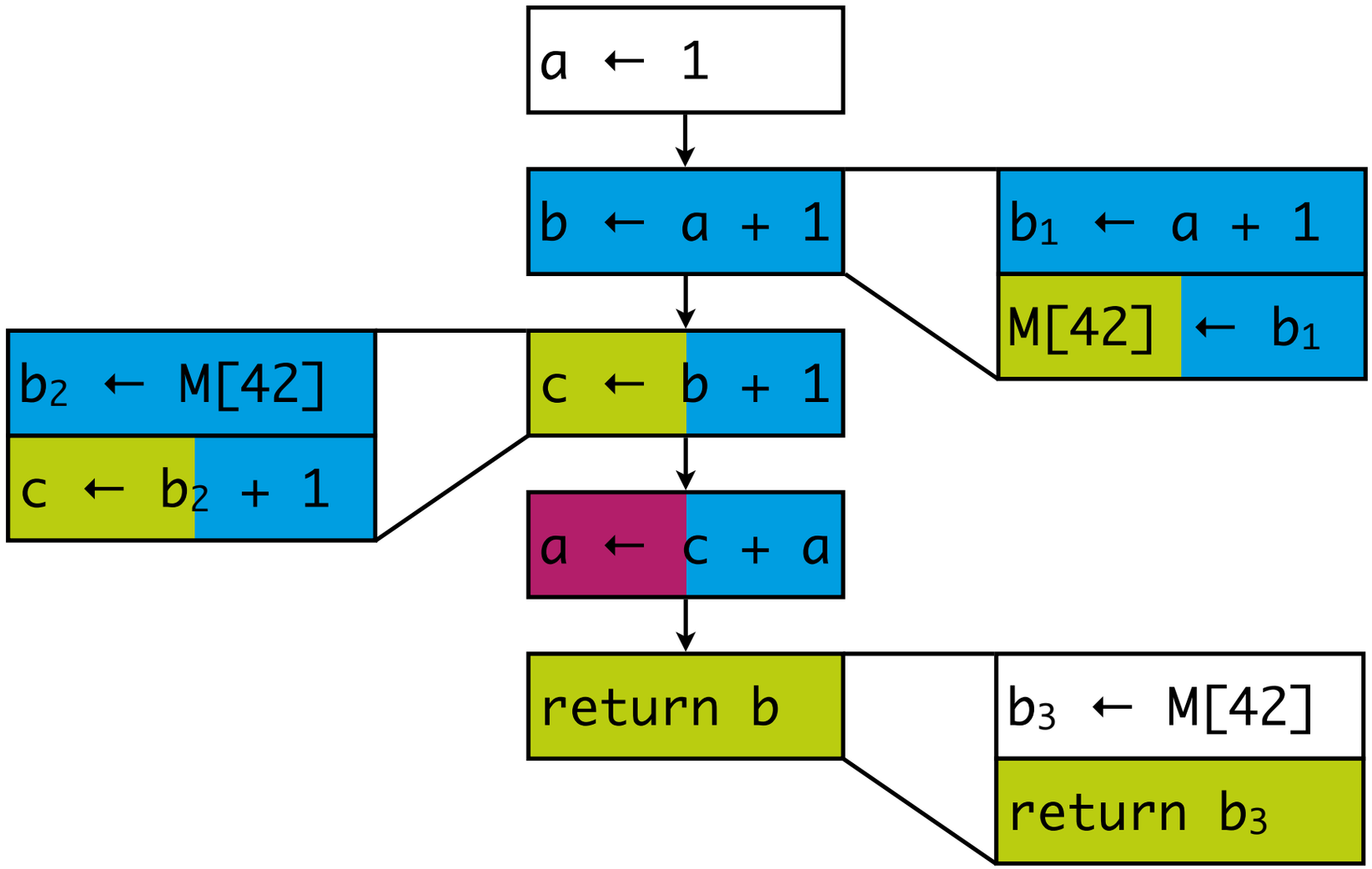
Spilling
example



Spilling
example



Spilling
example



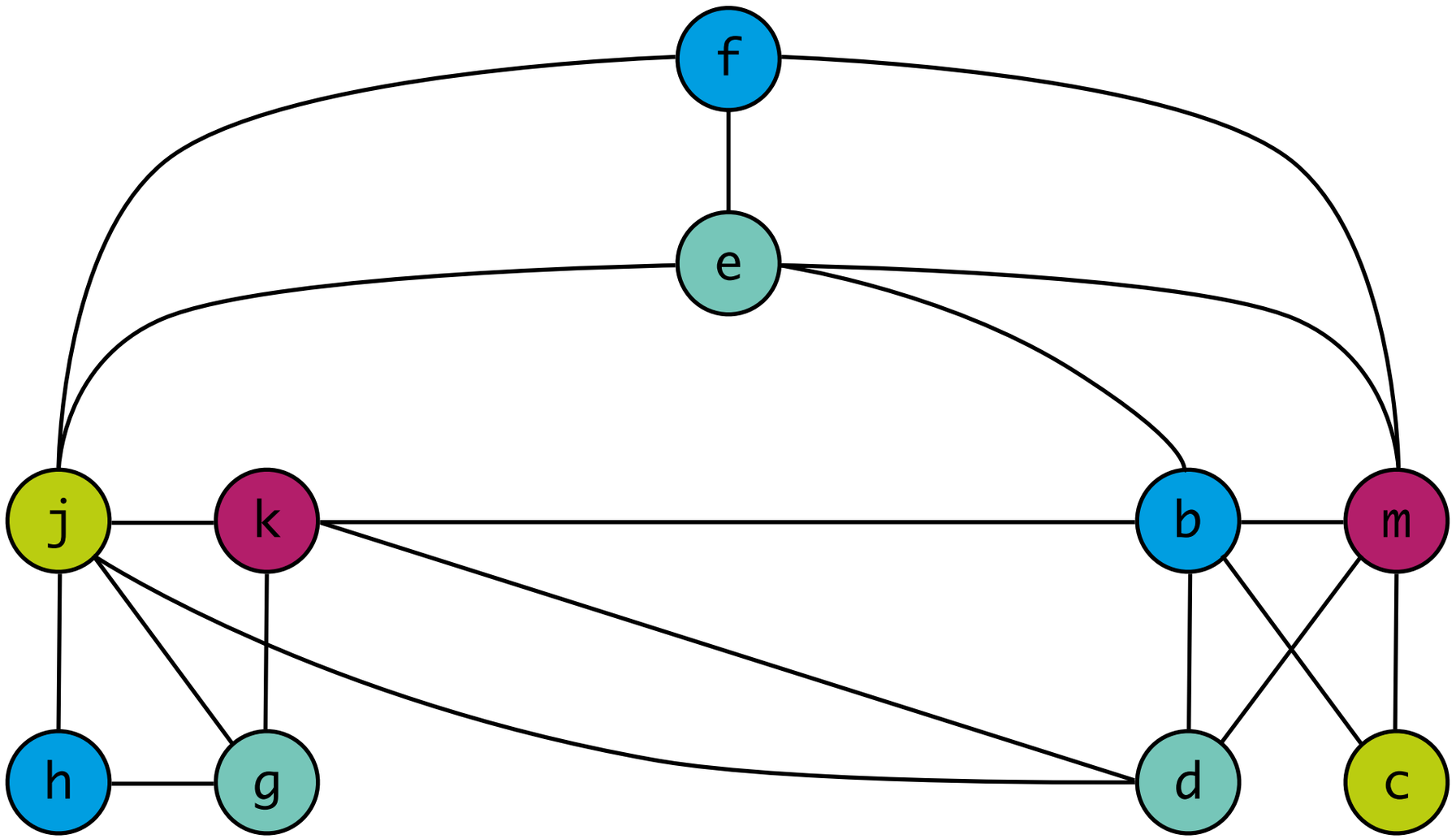
coffee break

III

coalescing

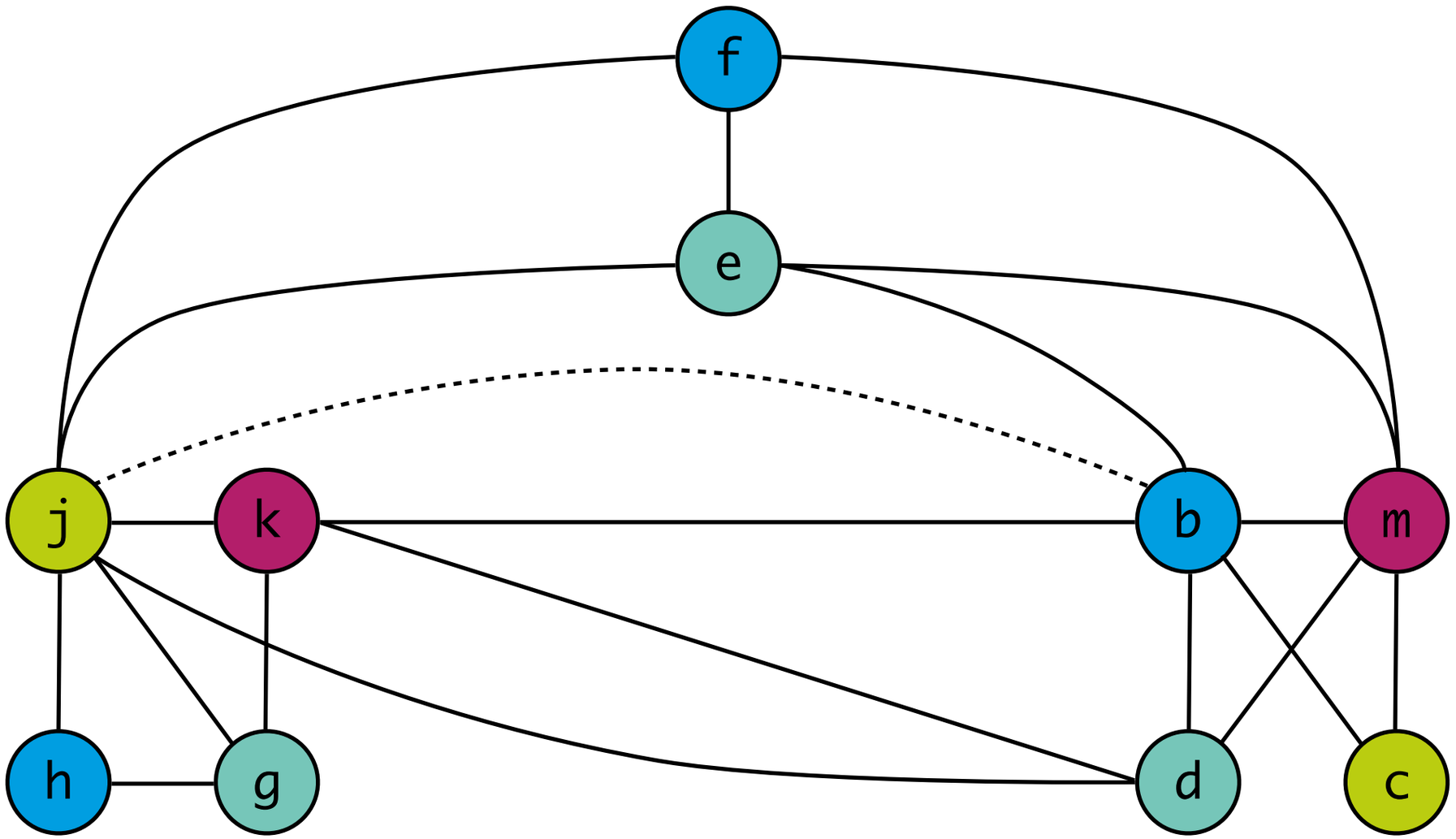
Recap: graph colouring

example



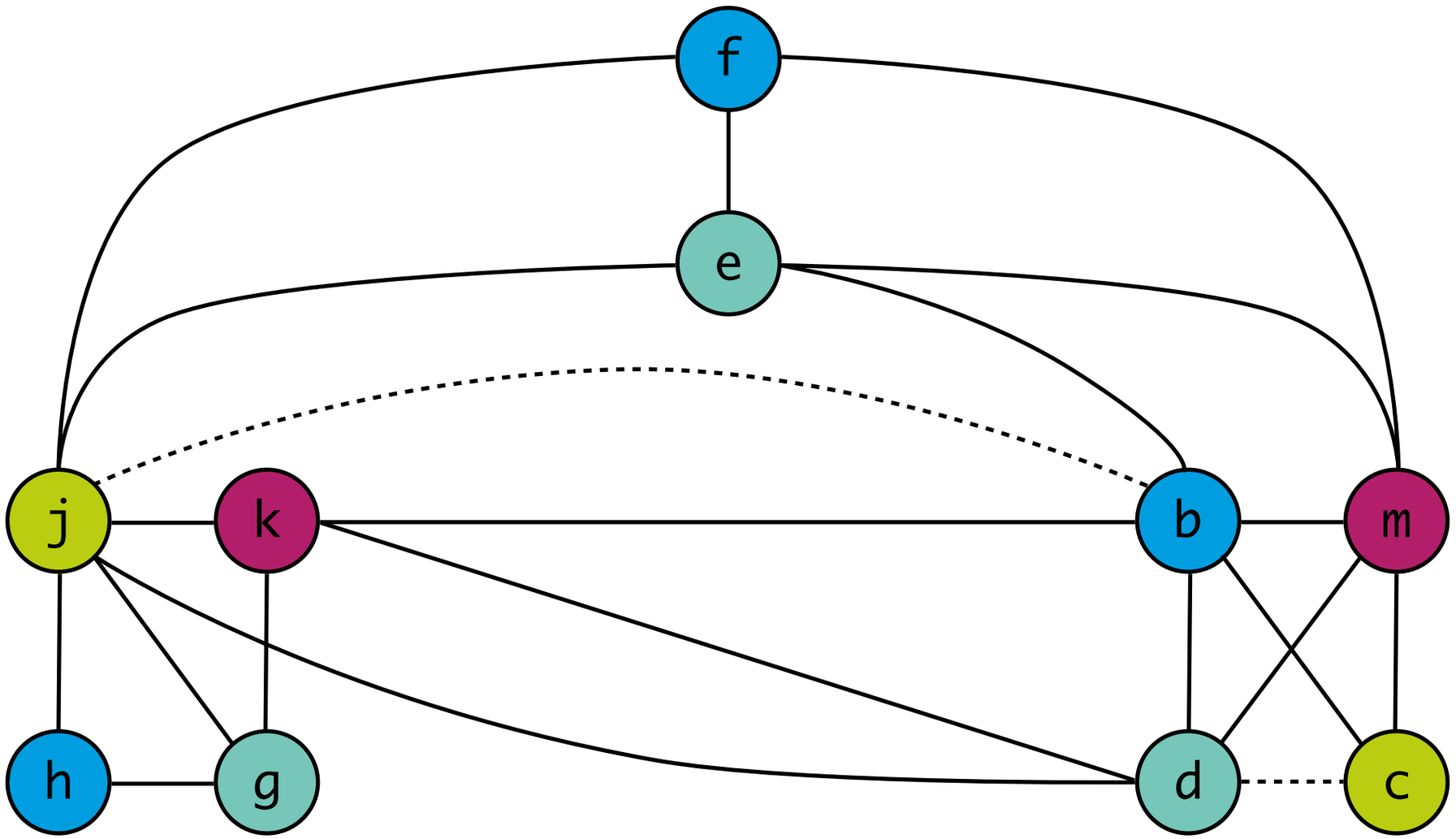
Recap: graph colouring

example



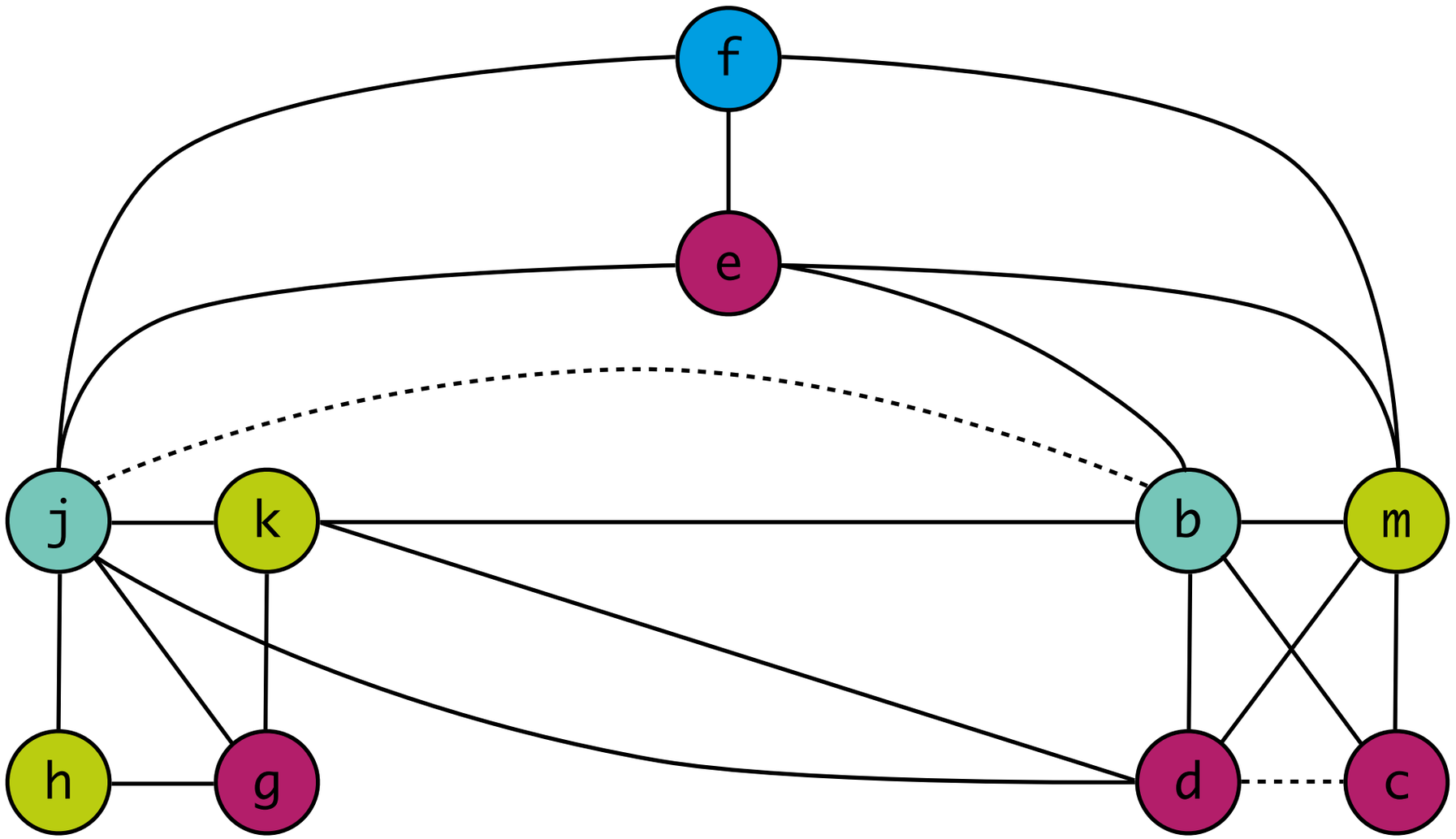
Recap: graph colouring

example



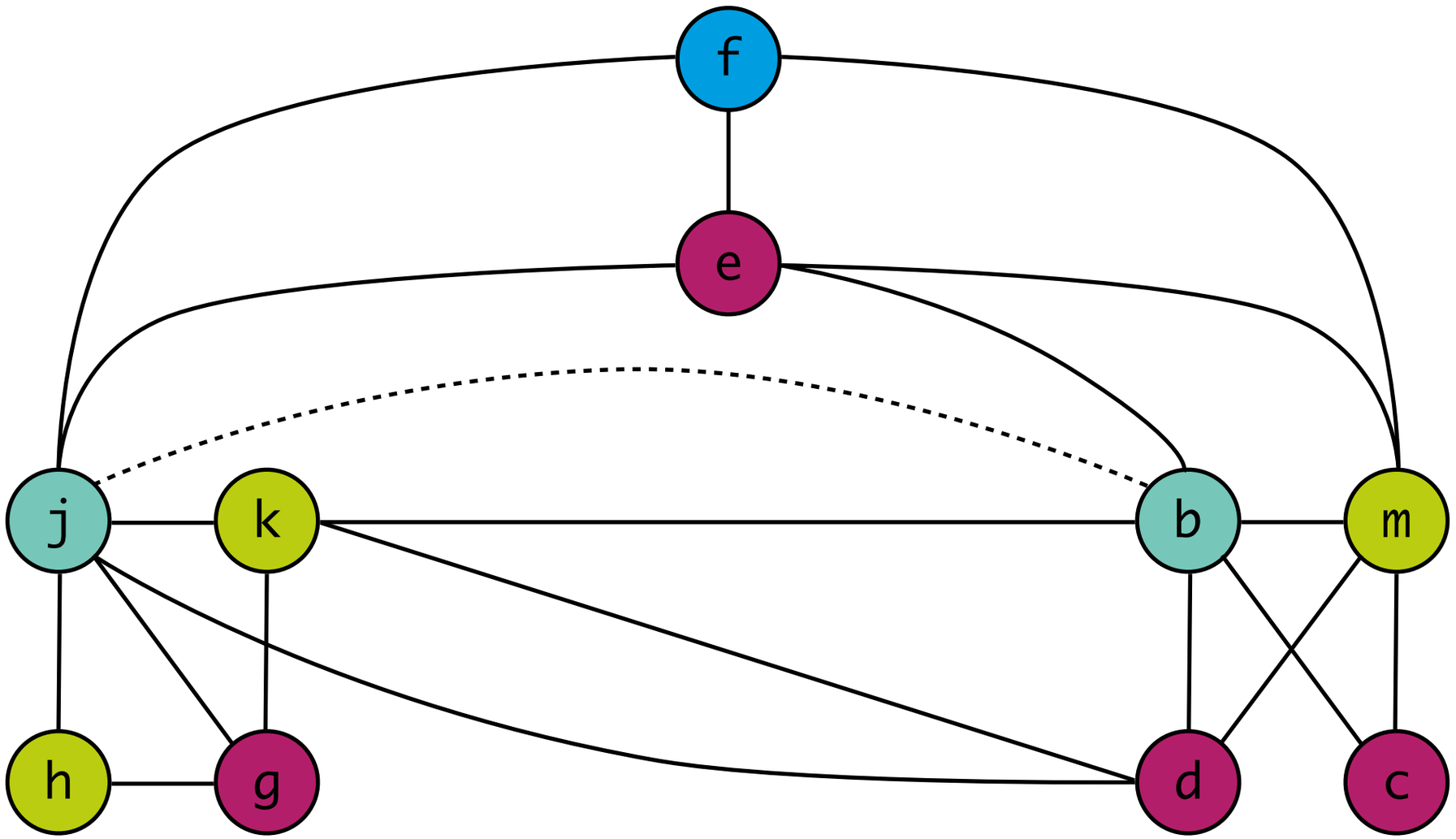
Coalescing

better solution



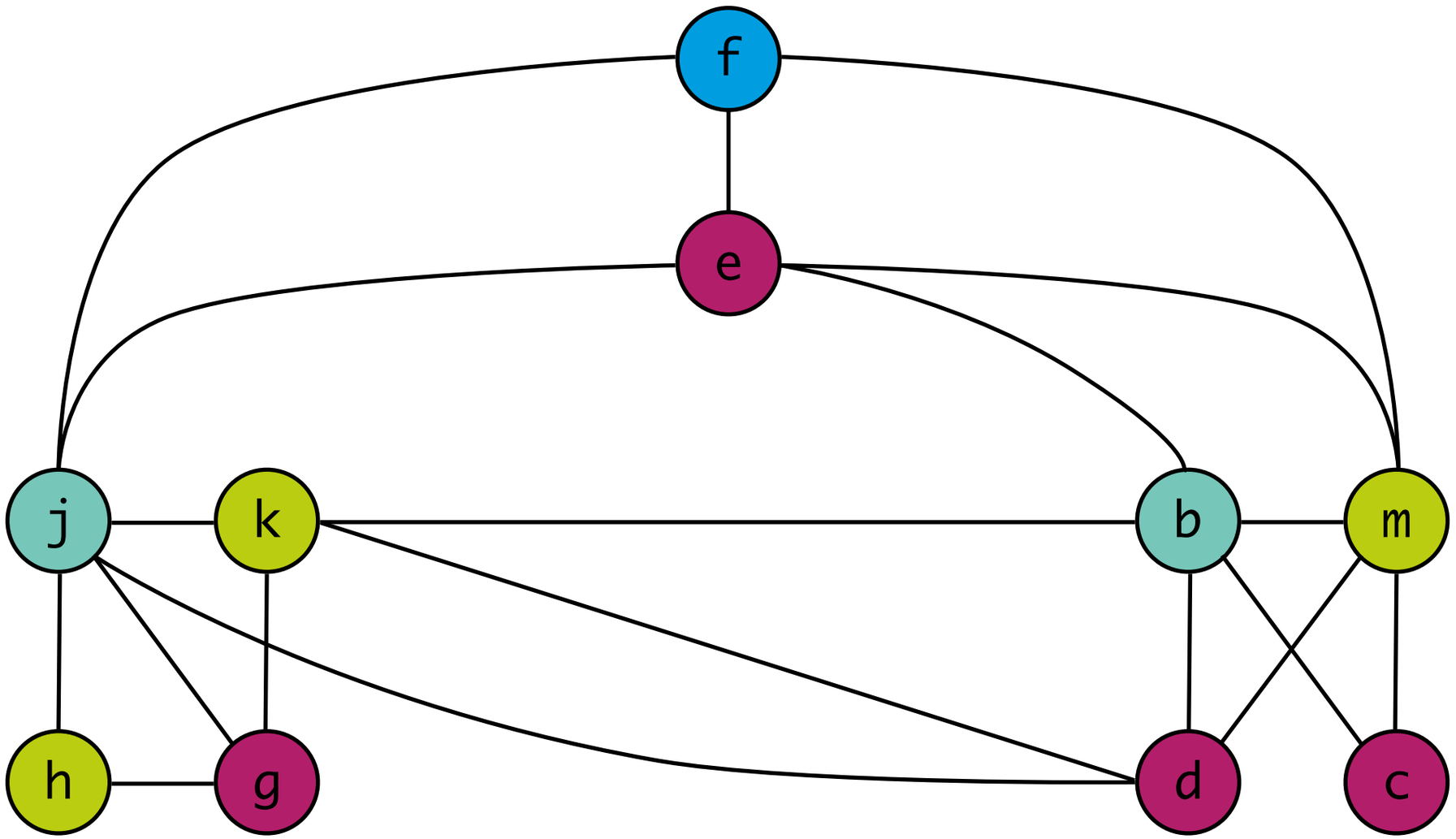
Coalescing

better solution



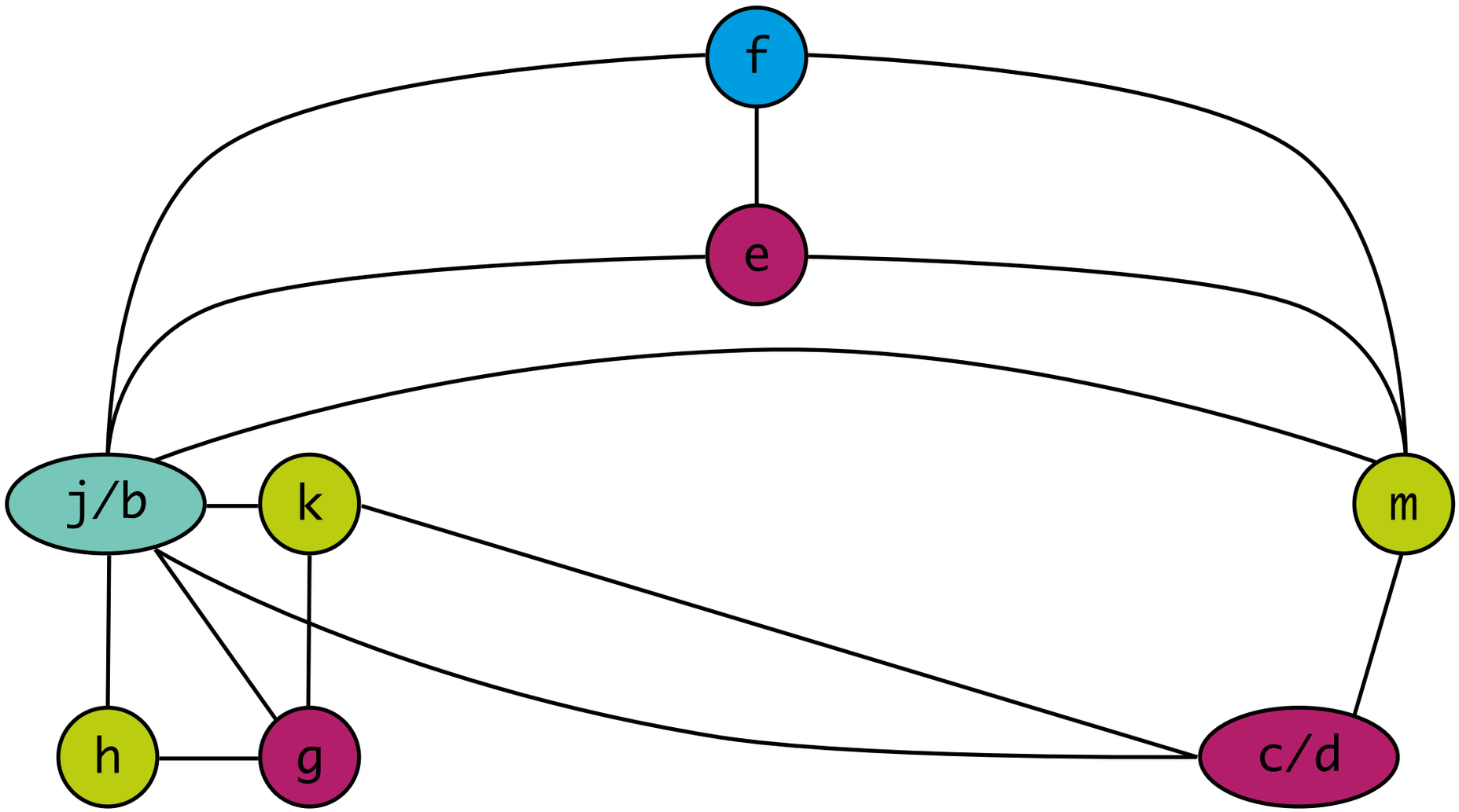
Coalescing

better solution



Coalescing

coalescing nodes



Coalescing

conservative strategies

Briggs

- a/b has fewer than k neighbours of significant degree
- nodes of insignificant degree and a/b can be simplified
- remaining graph is colourable

George

- all neighbours of a of significant degree interfere also with b
- neighbours of a of insignificant degree can be simplified
- subgraph of original graph is colourable

Graph colouring

steps

simplify

remove non-move-related node of insignificant degree

coalesce

freeze

turn move-related node of insignificant degree into non-move-related

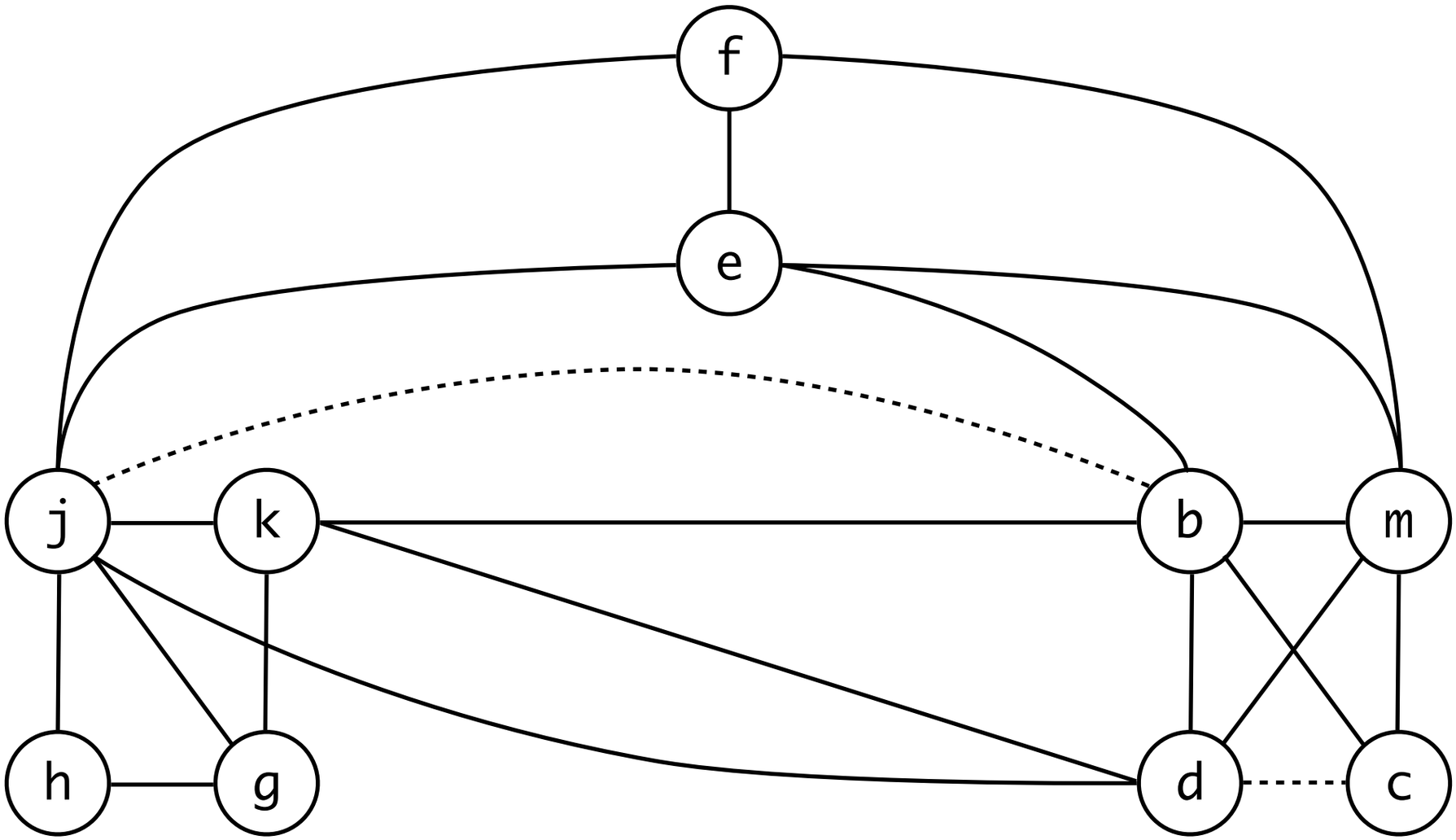
spill

select

start over

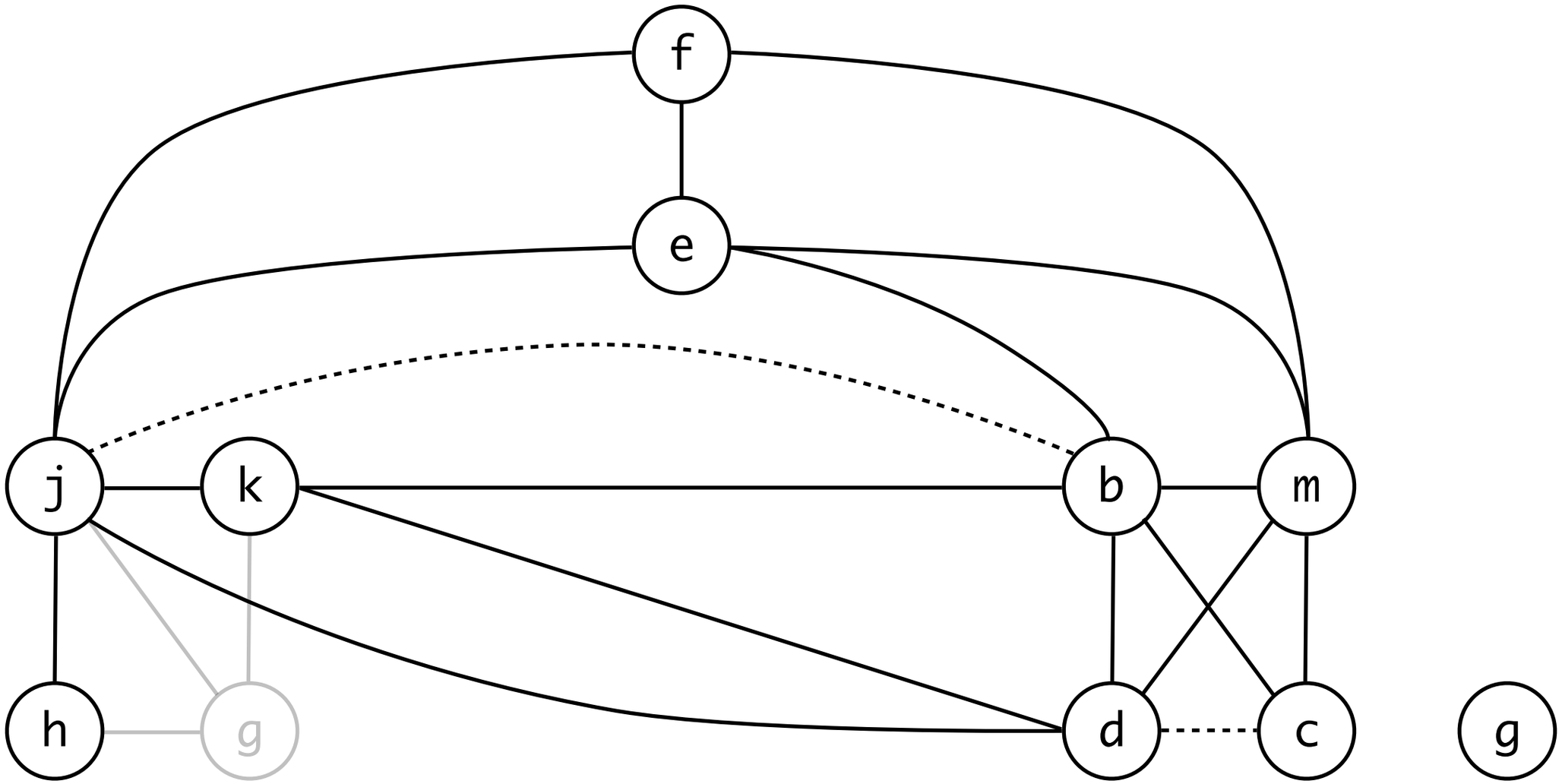
Coalescing

example



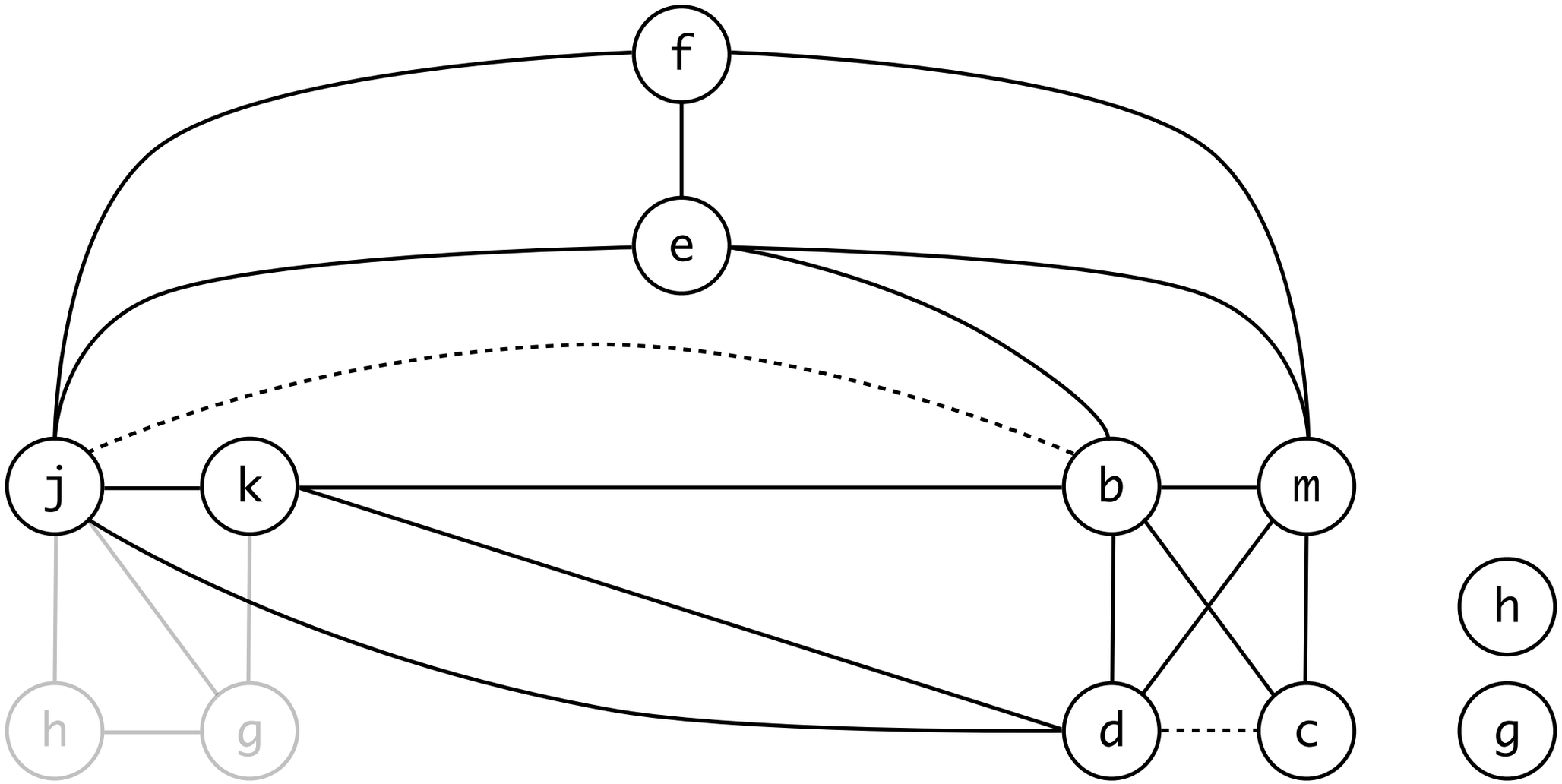
Coalescing

example



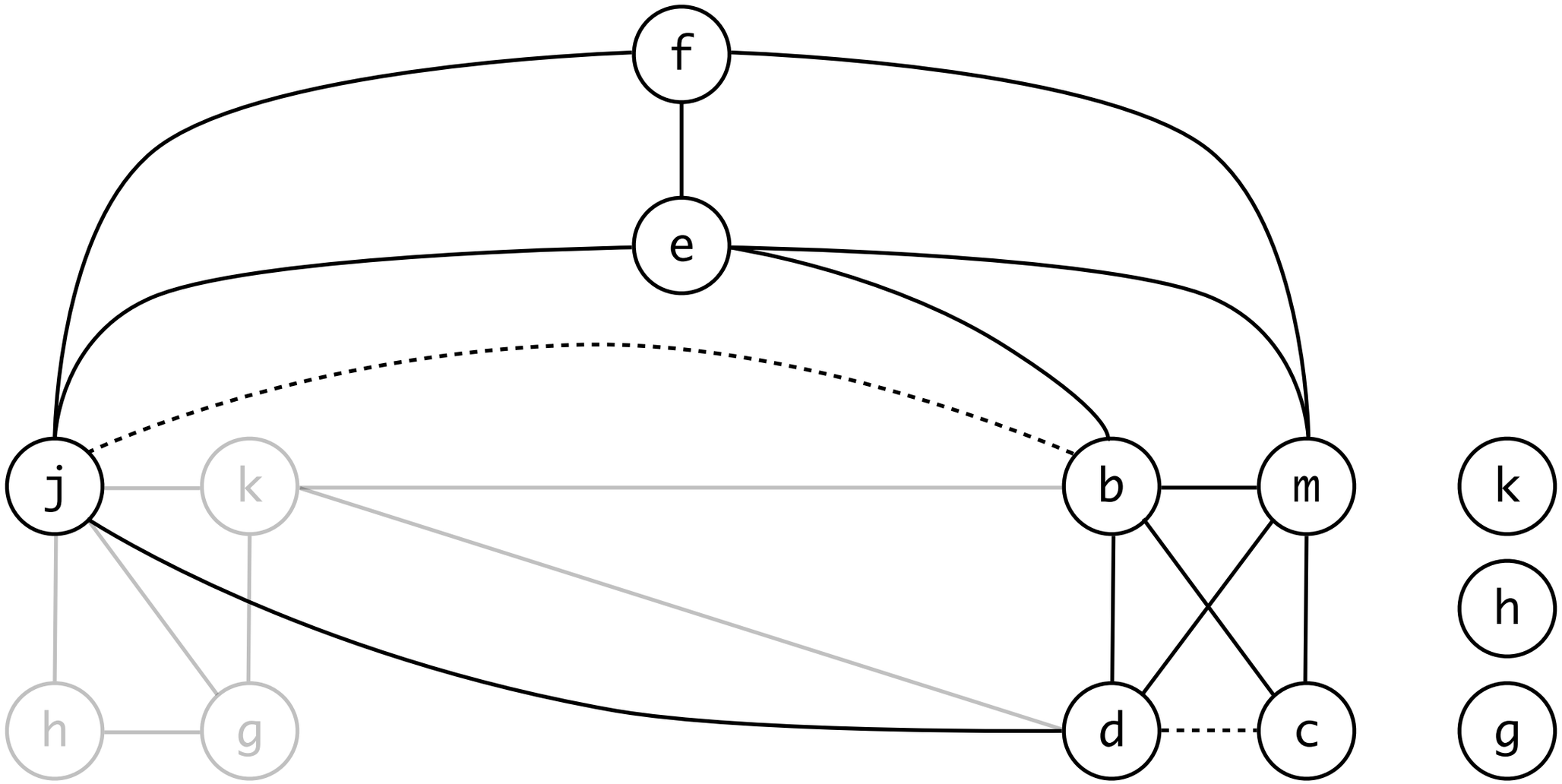
Coalescing

example



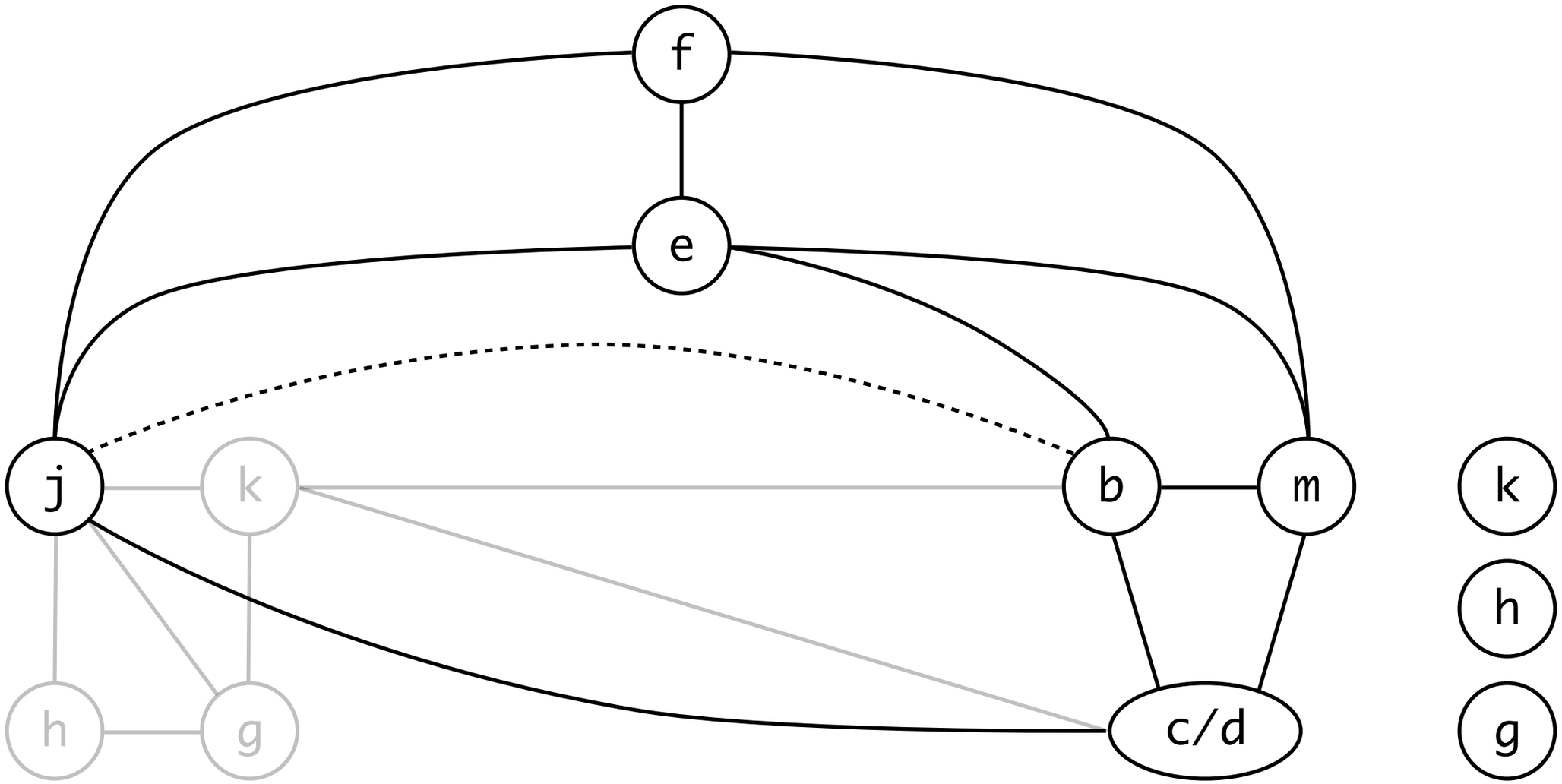
Coalescing

example



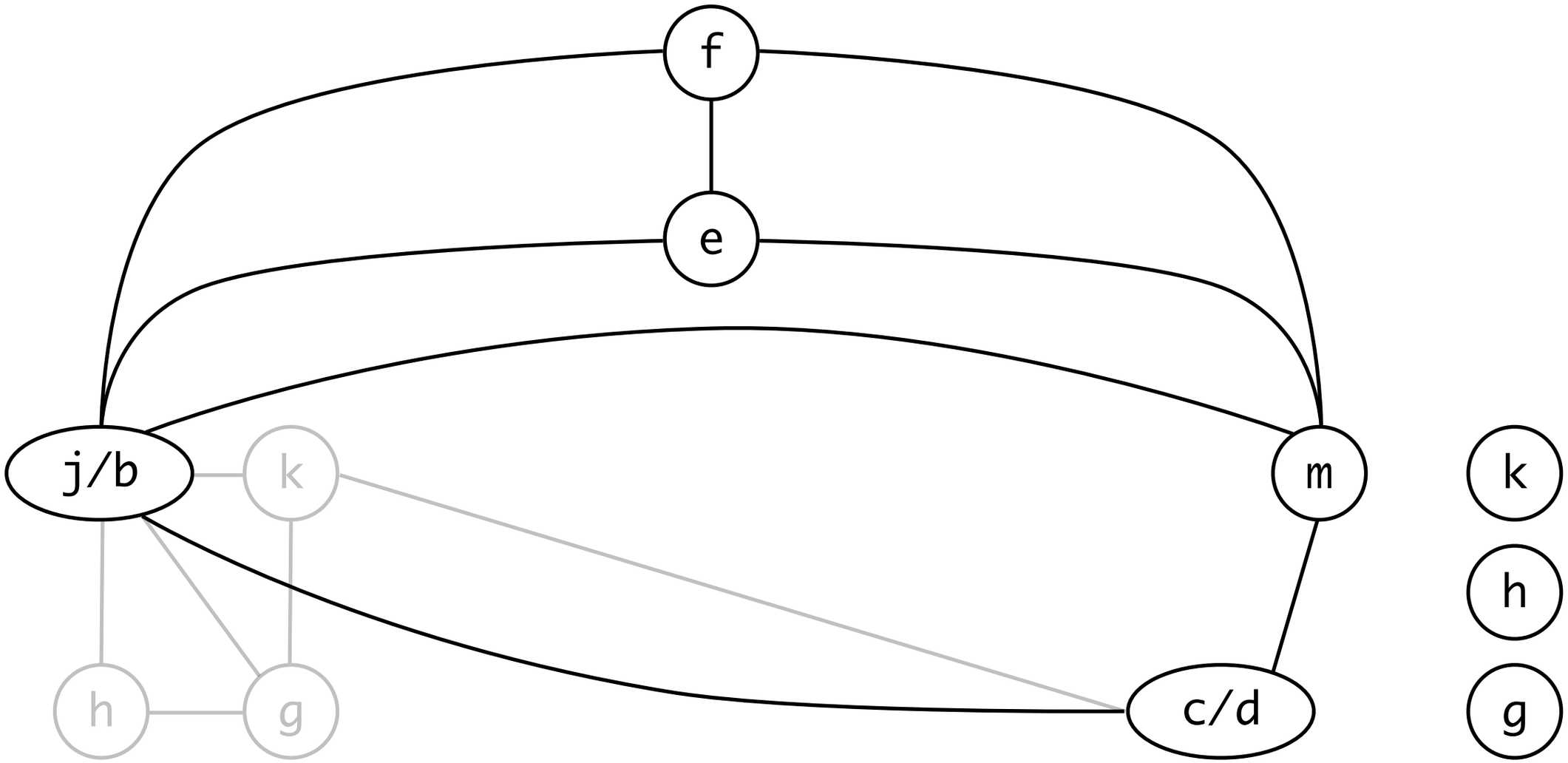
Coalescing

example



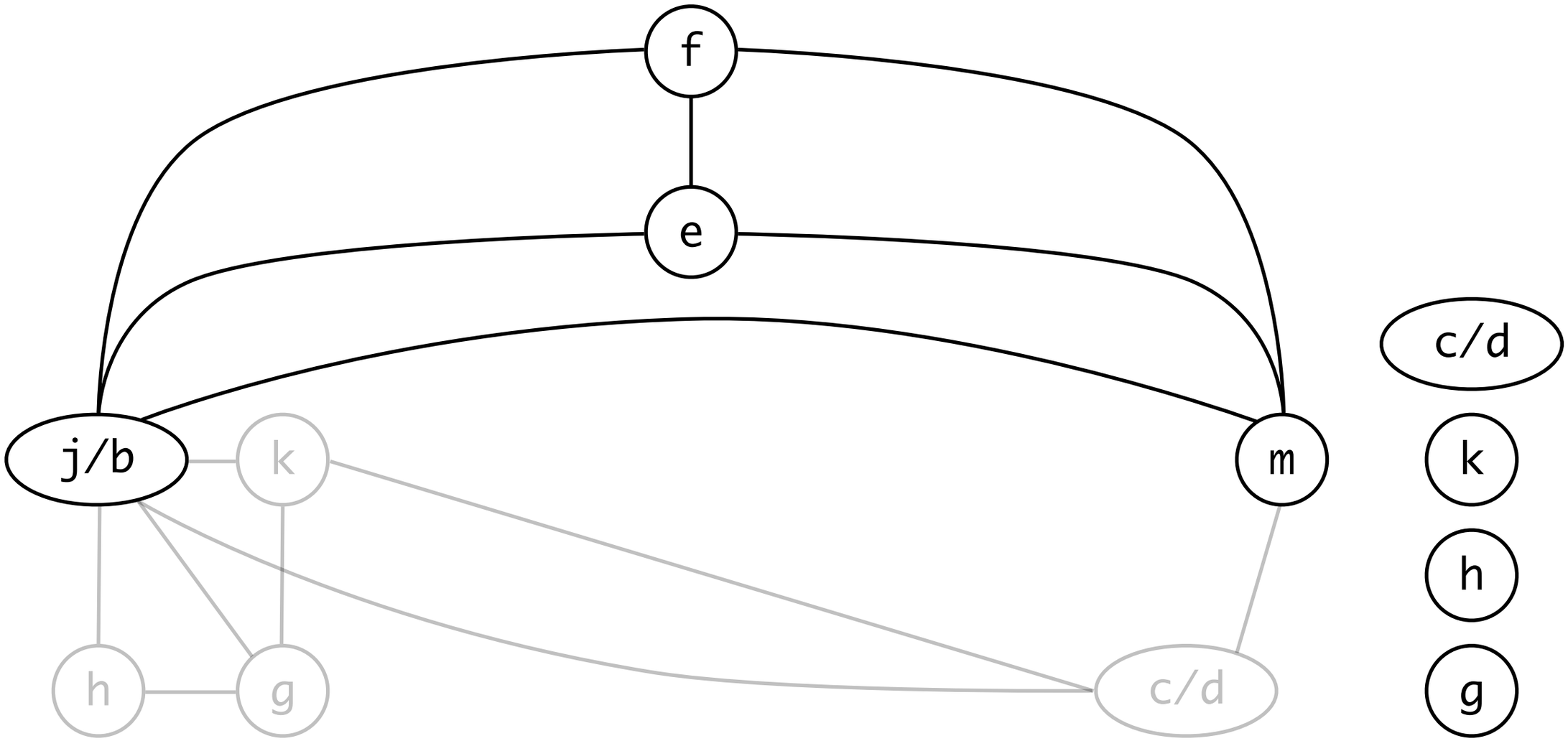
Coalescing

example



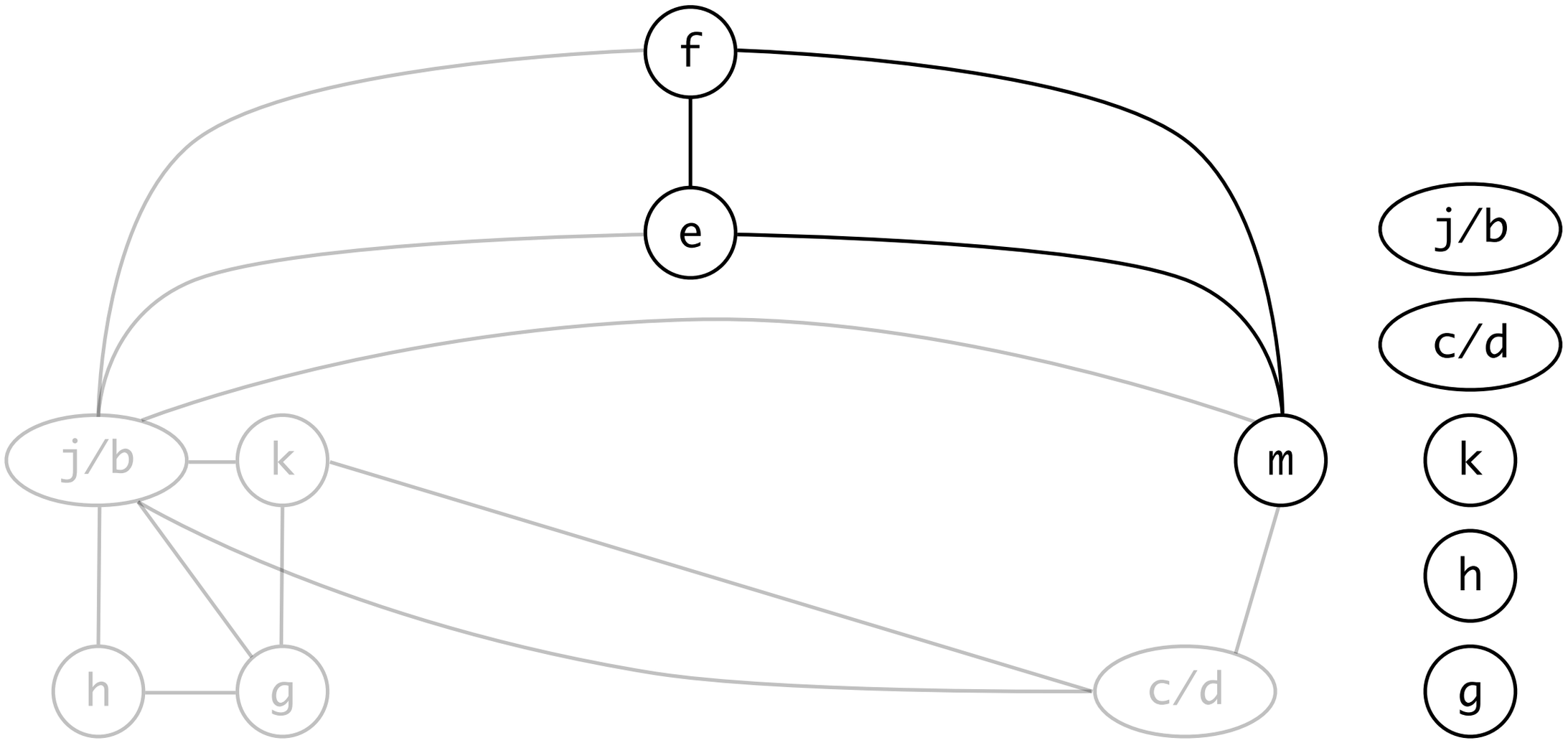
Coalescing

example



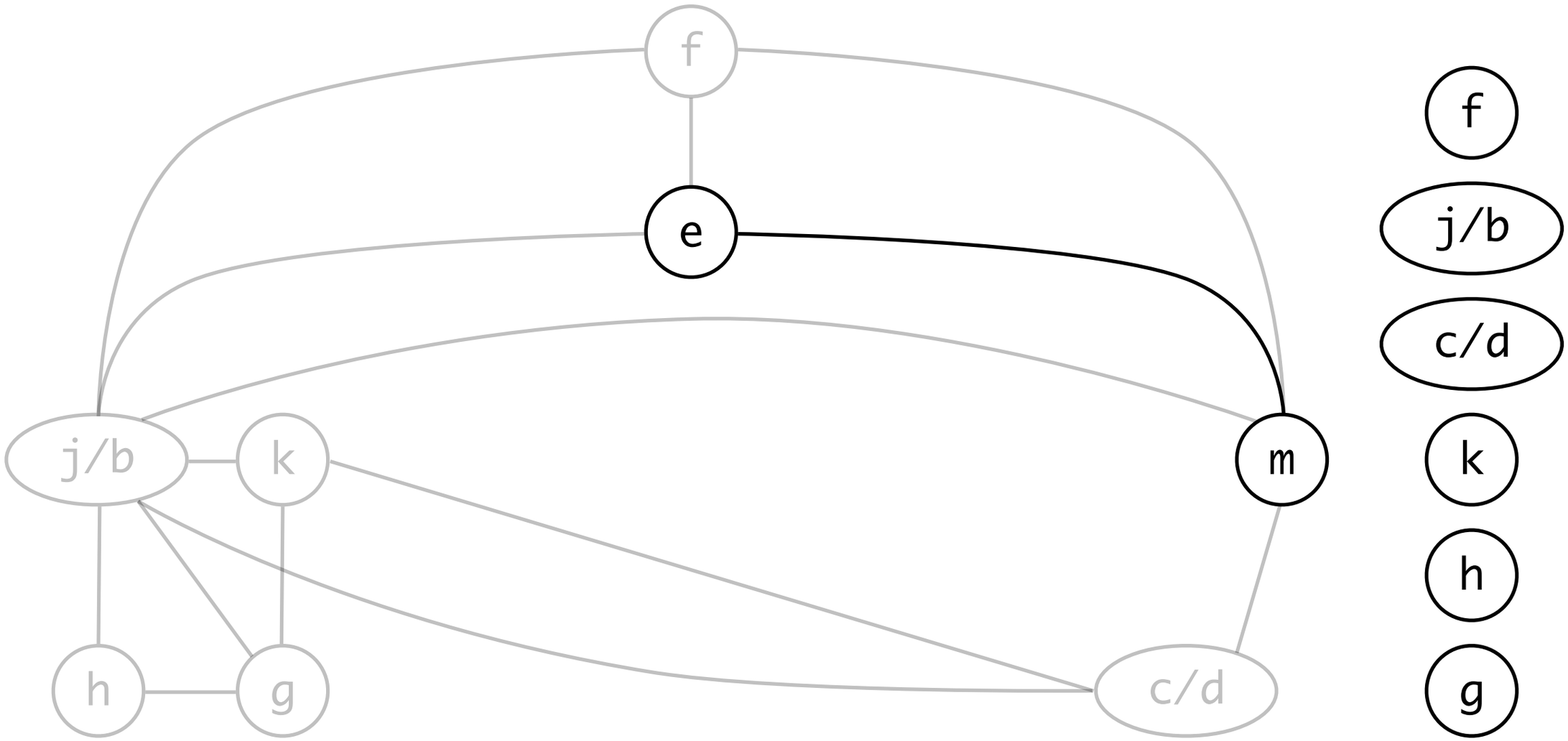
Coalescing

example



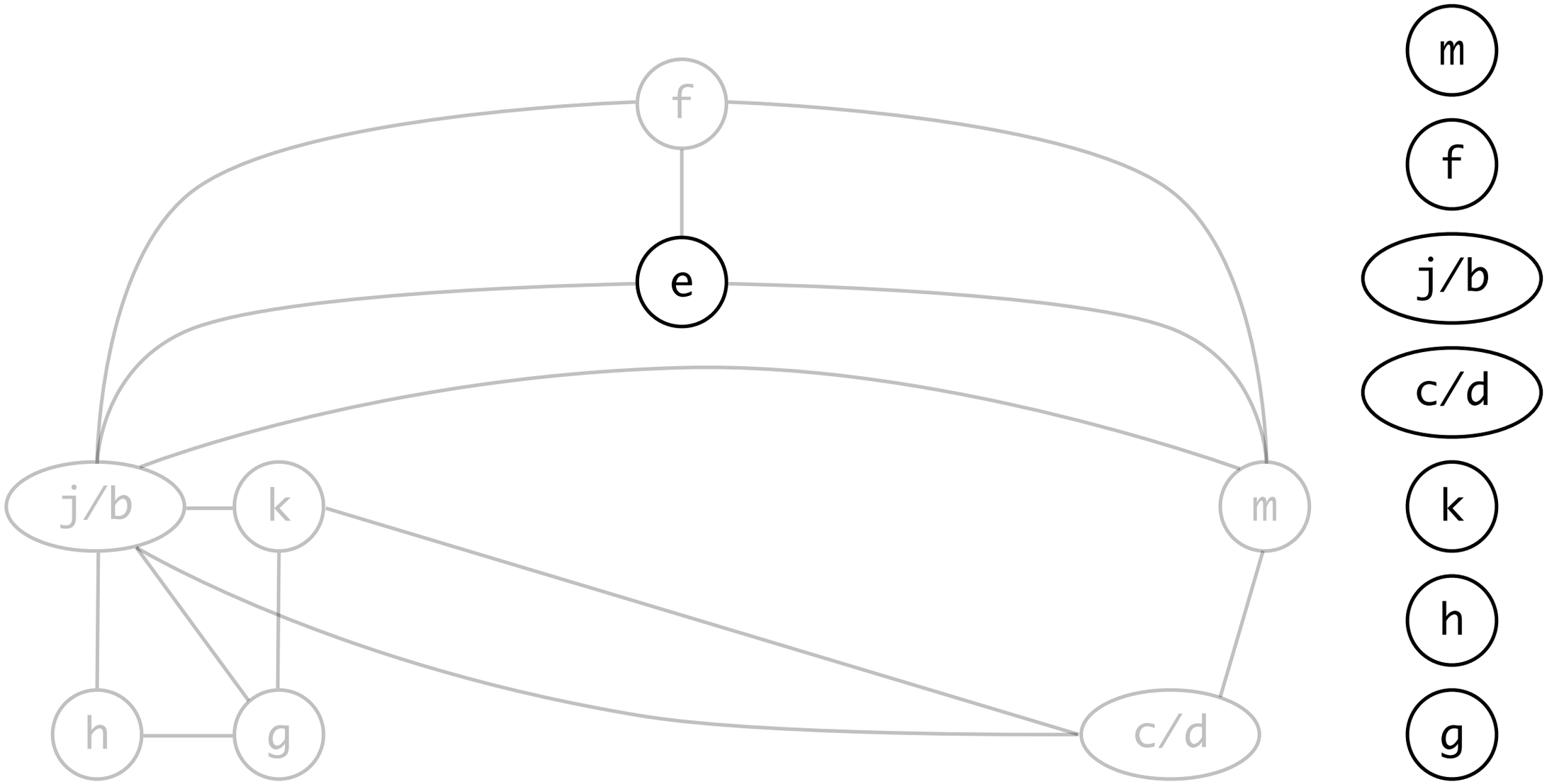
Coalescing

example



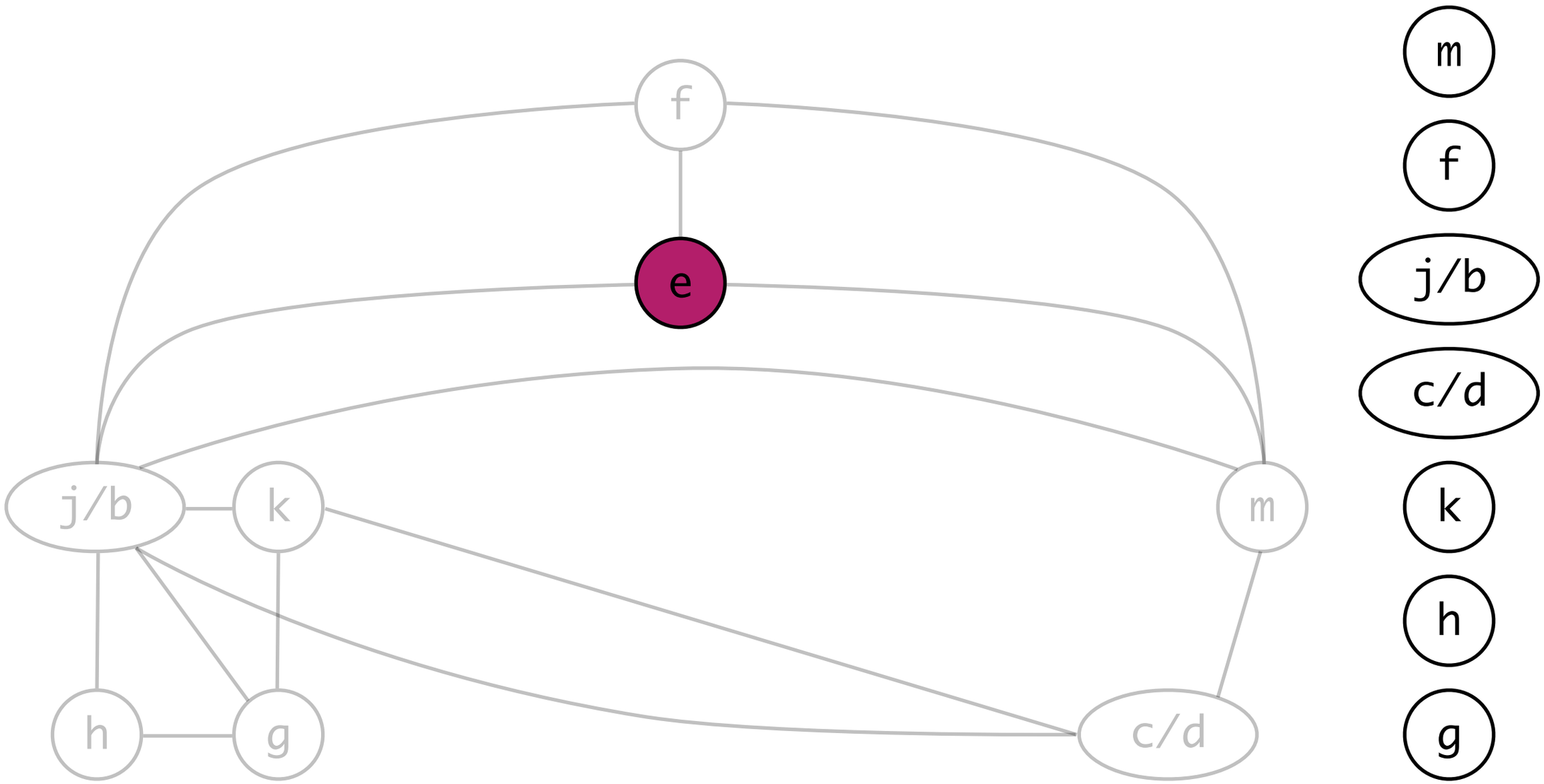
Coalescing

example



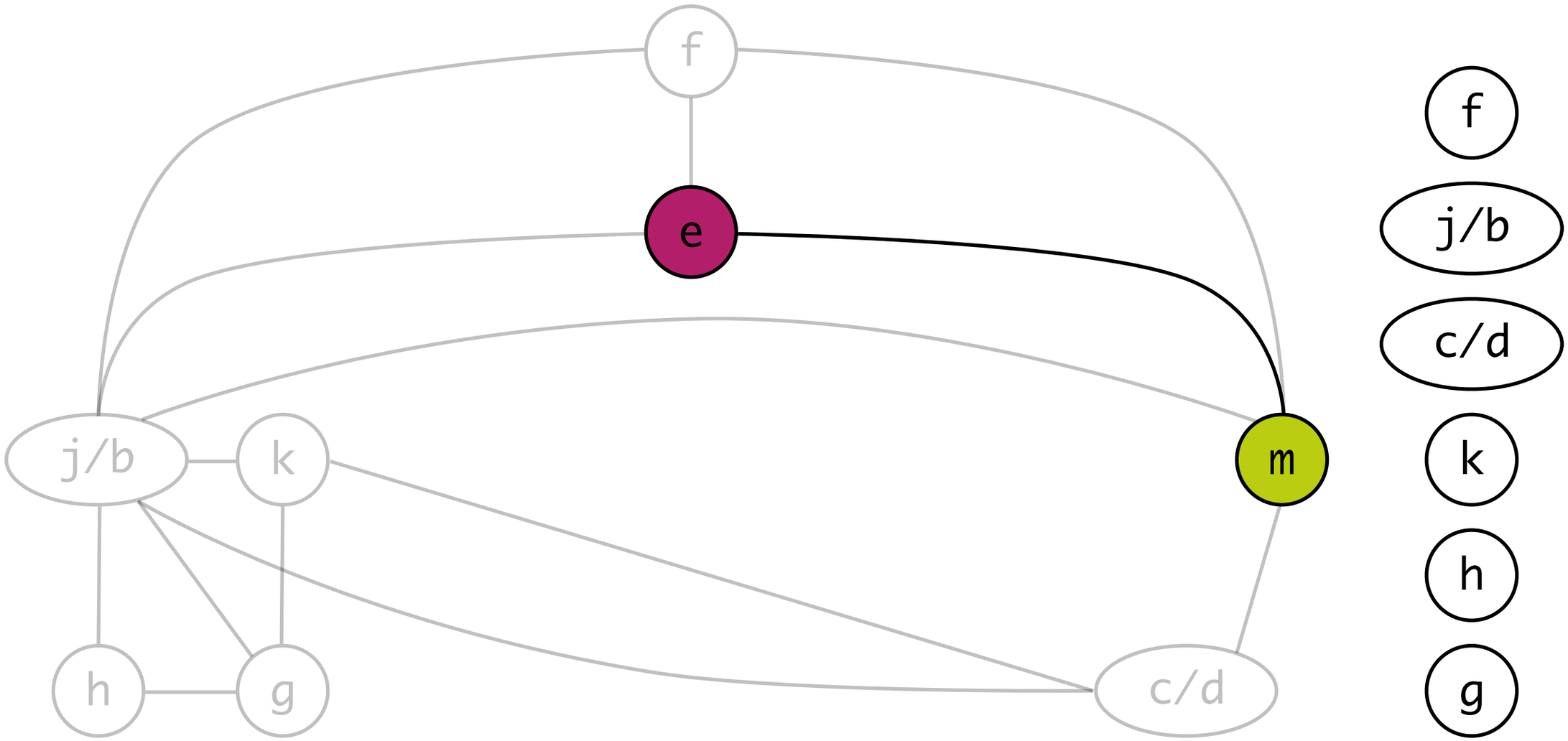
Coalescing

example



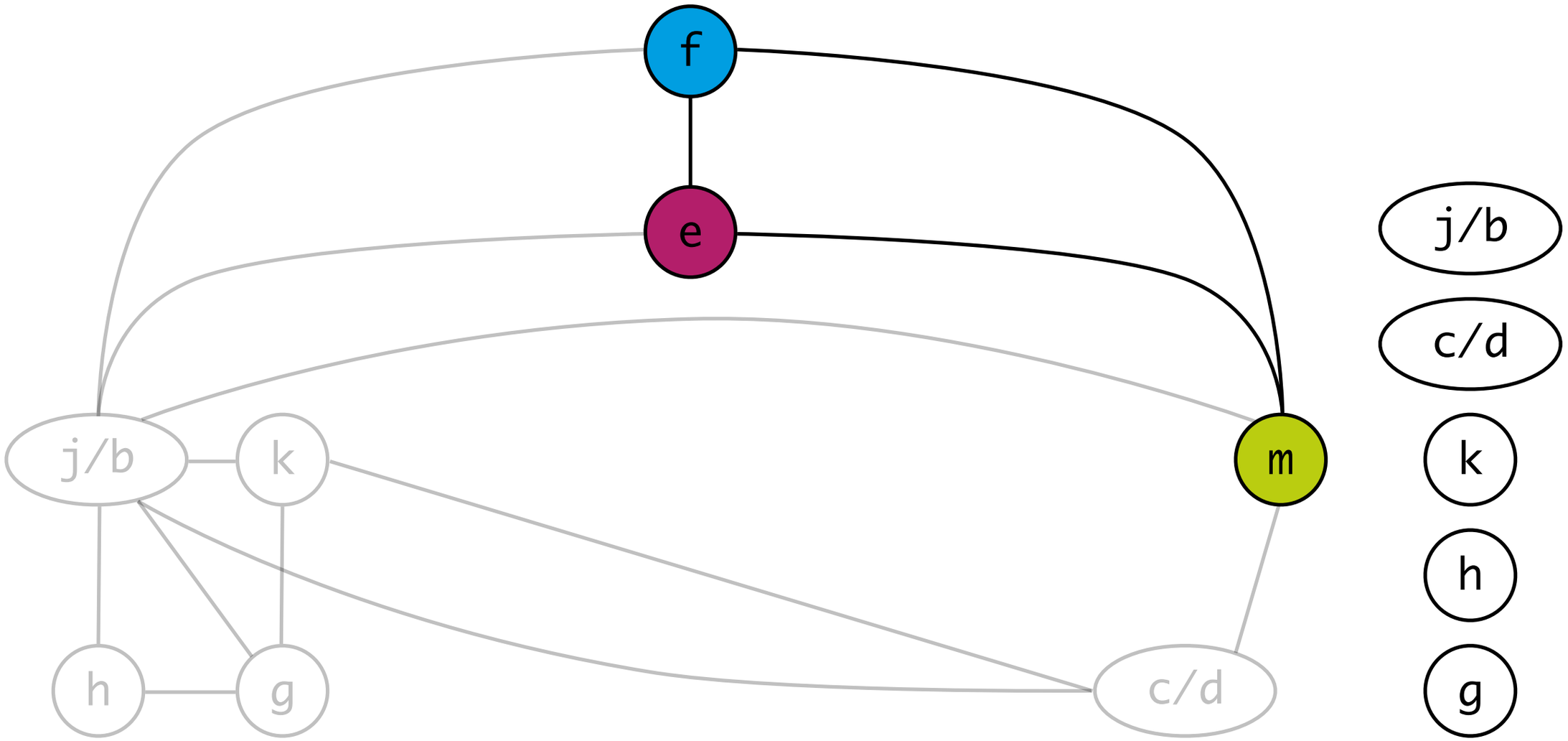
Coalescing

example



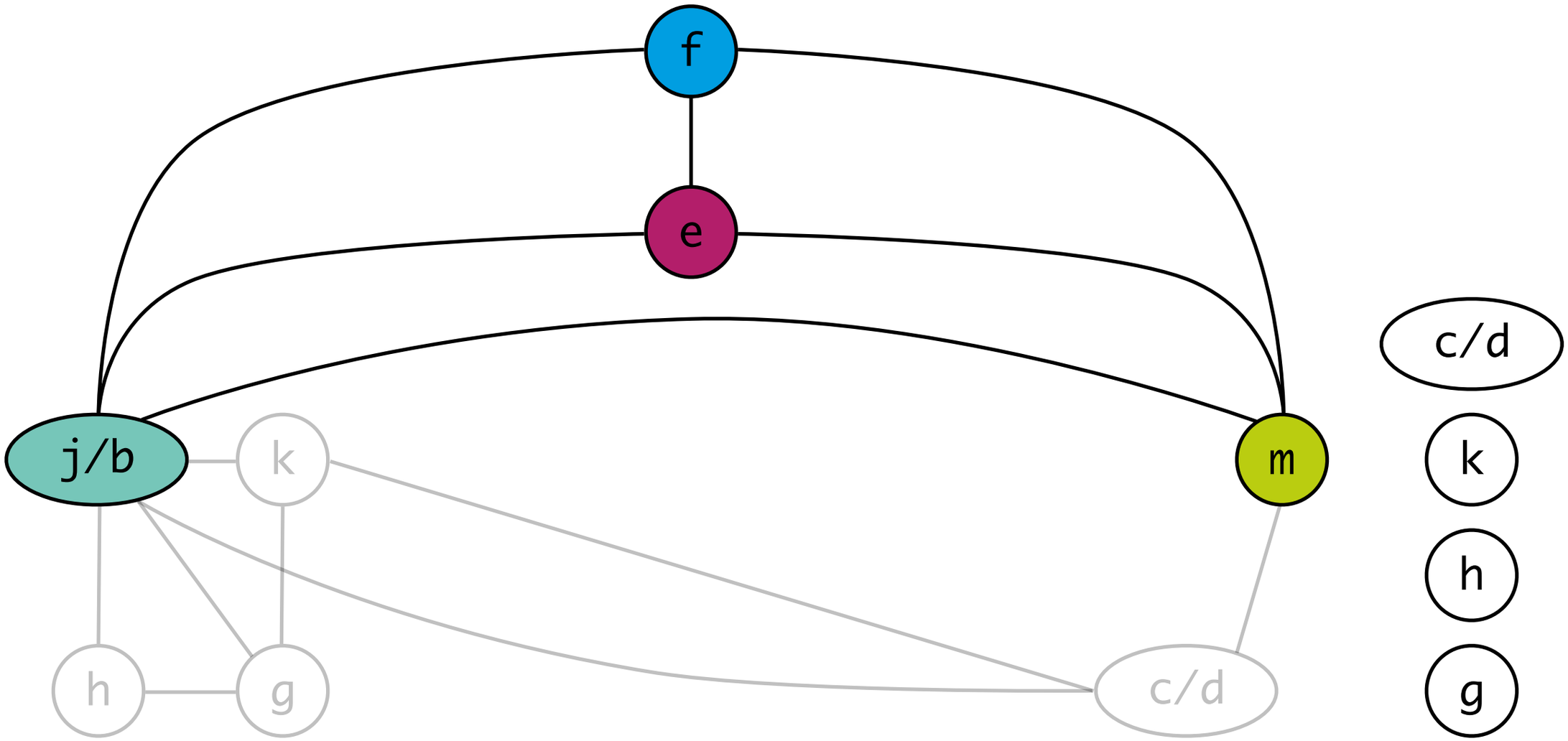
Coalescing

example



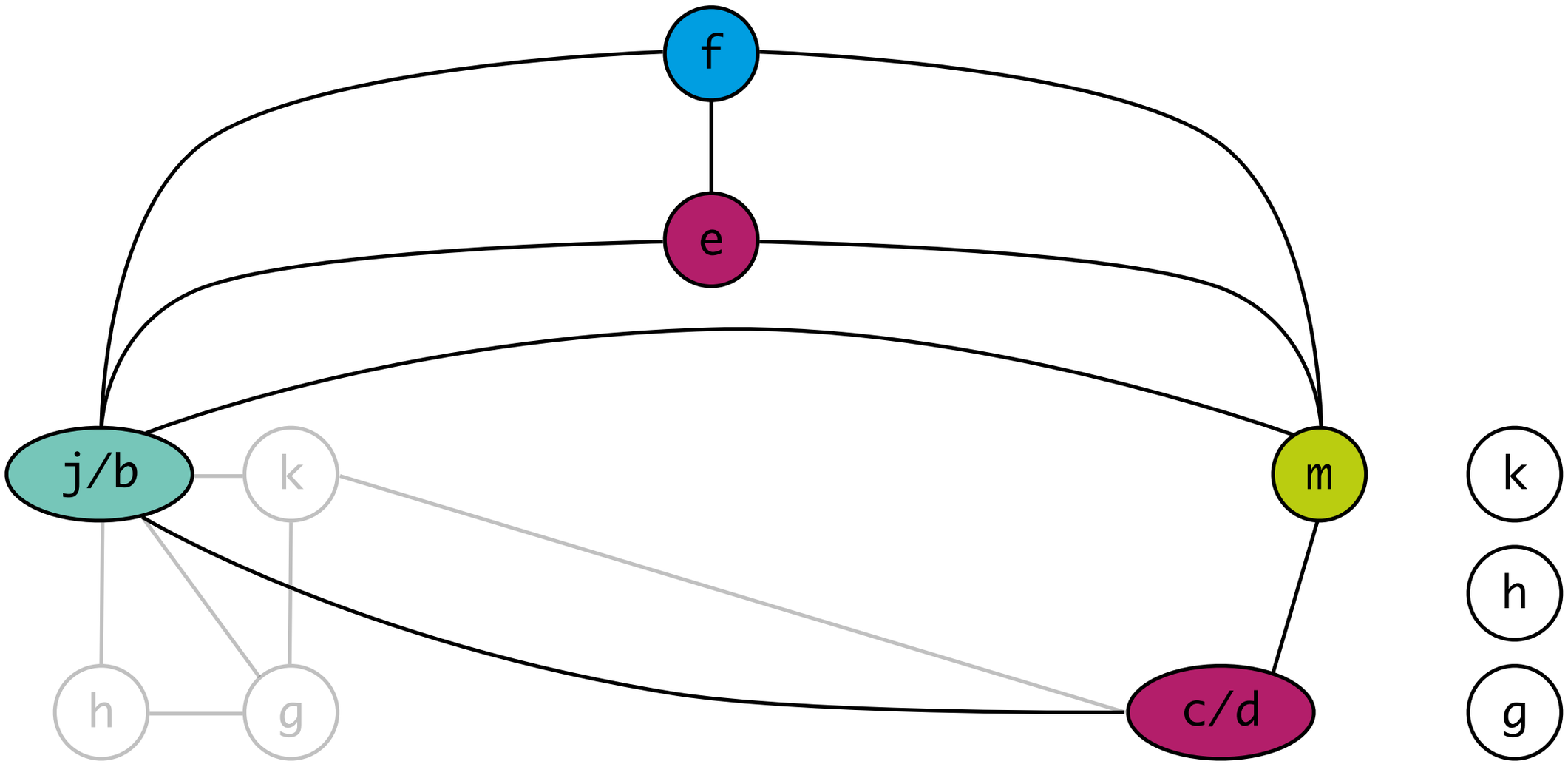
Coalescing

example



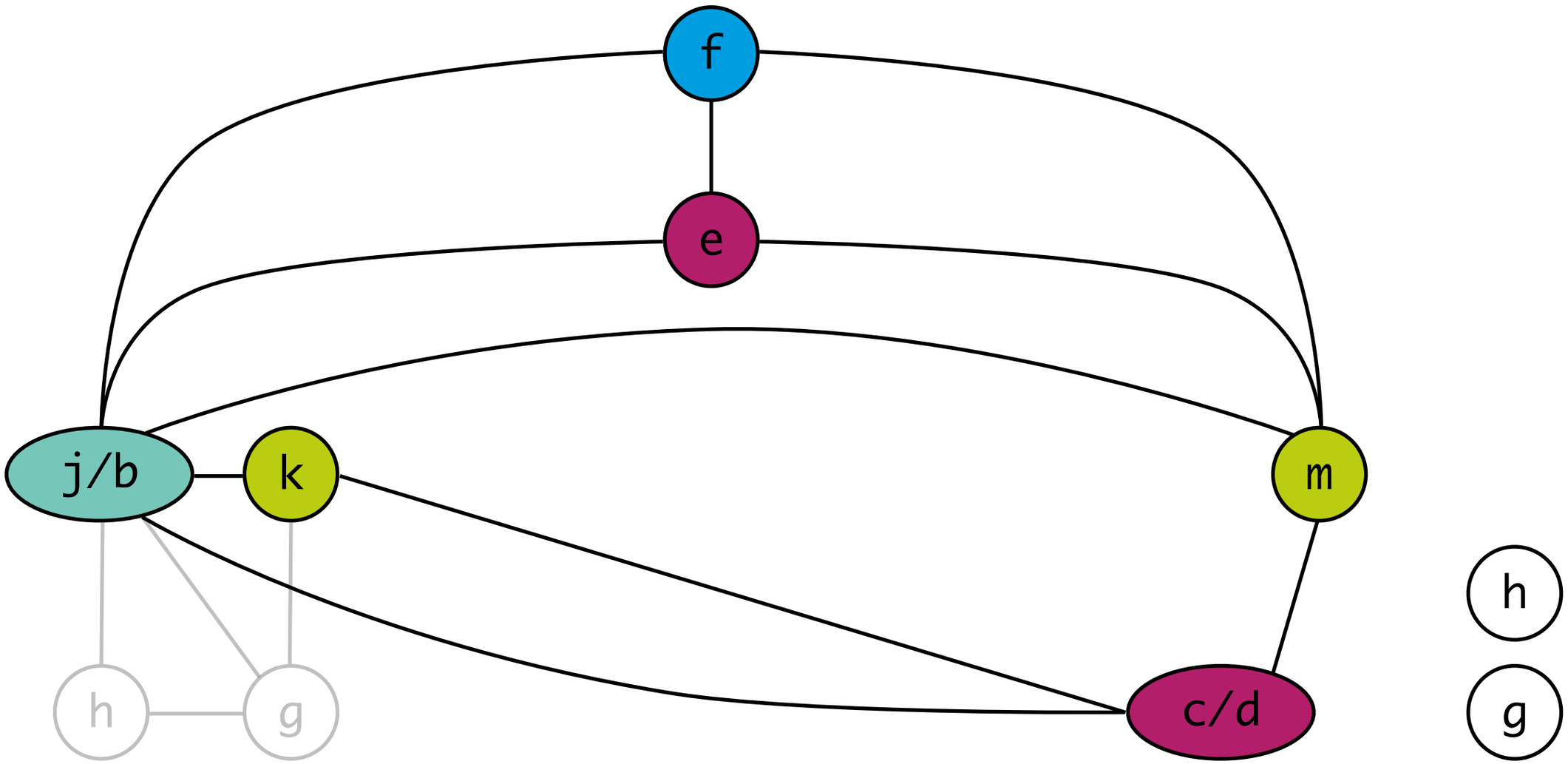
Coalescing

example



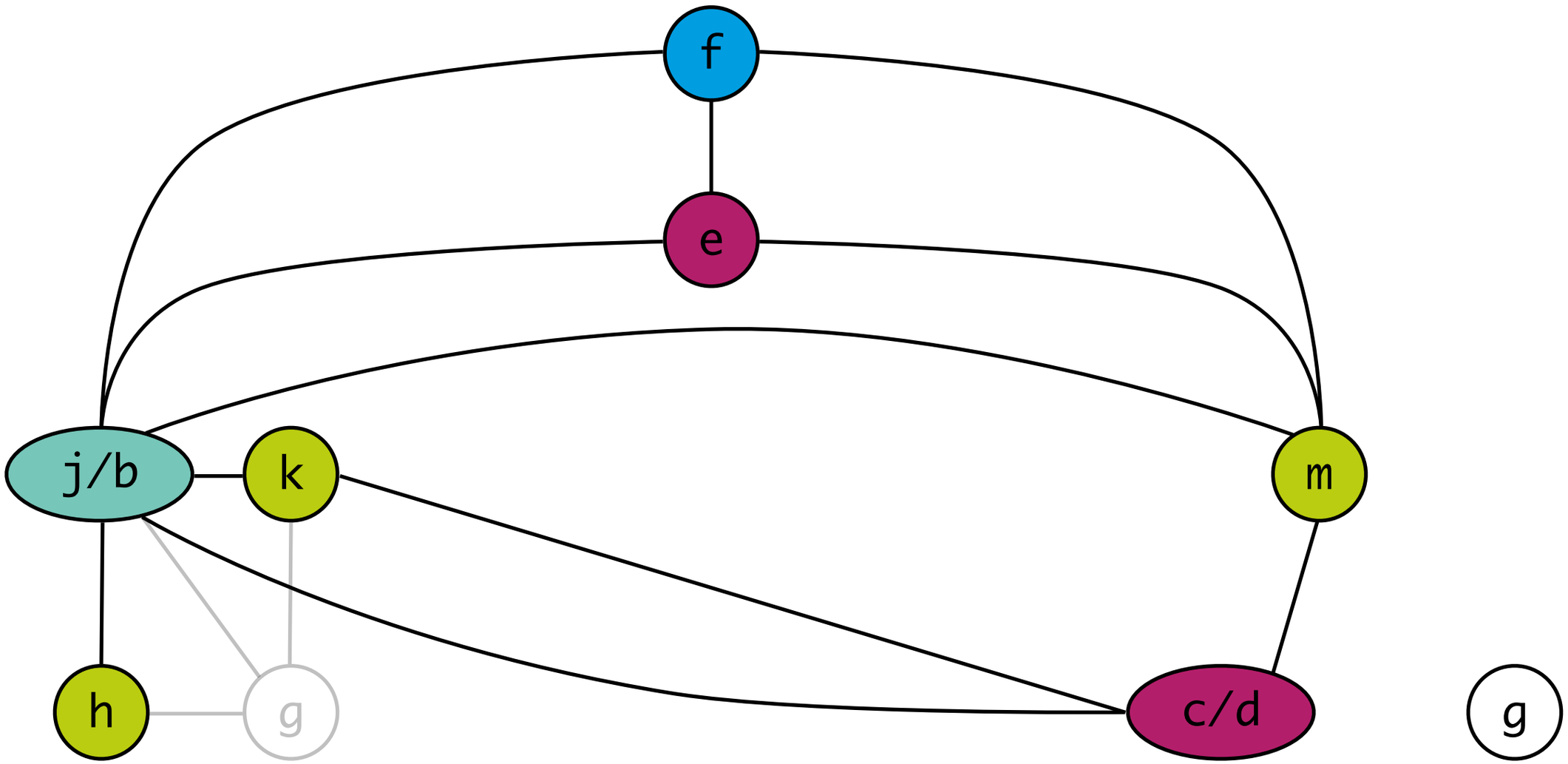
Coalescing

example



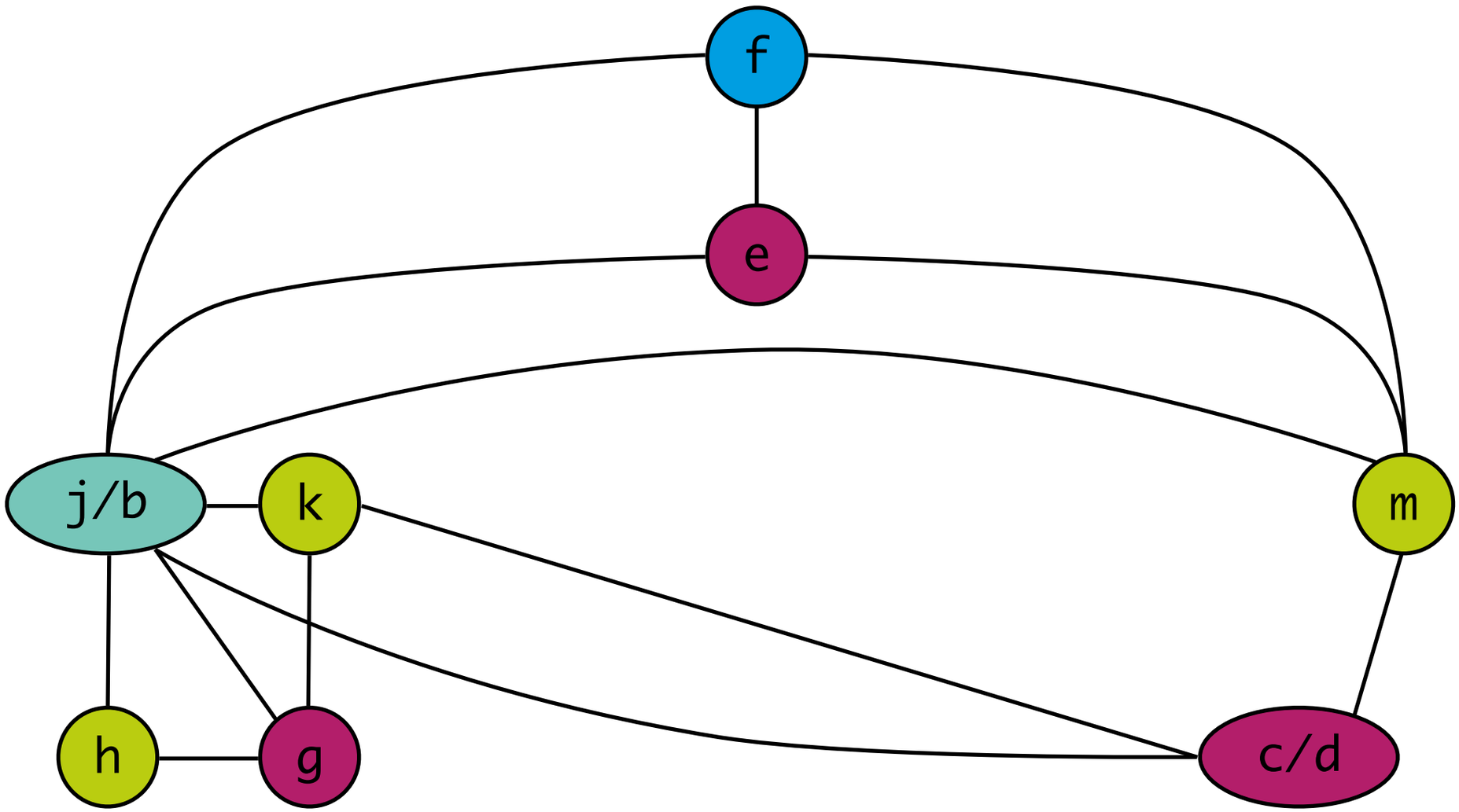
Coalescing

example



Coalescing

coalescing nodes



IV

pre-coloured nodes

Recap: Calling Conventions

CDECL

caller

- push parameters right-to-left on the stack
- clean-up stack after call

```
push 21
push 42
call _f
add  ESP 8
```

callee

- save old BP
- initialise new BP
- save registers
- return result in AX
- restore registers
- restore BP

```
push EBP
mov  EBP ESP
mov  EAX [EBP + 8]
mov  EDX [EBP + 12]
add  EAX EDX
pop  EBP
ret
```

Recap: Calling Conventions
STDCALL

caller

- push parameters right-to-left on the stack

```
push 21
push 42
call _f@8
```

callee

- save old BP
- initialise new BP
- save registers
- return result in AX
- restore registers
- restore BP

```
push EBP
mov  EBP ESP
mov  EAX [EBP + 8]
mov  EDX [EBP + 12]
add  EAX EDX
pop  EBP
ret  8
```

Recap: Calling Conventions
FASTCALL

caller

- passes parameters in registers
- pushes additional parameters right-to-left on the stack
- cleans up the stack

```
mov   ECX 21
mov   EDX 42
call  @f@8
```

callee

- save old BP, initialise new BP
- save registers
- return result in AX
- restore registers
- restore BP

```
push EBP
mov  EBP ESP
mov  EAX ECX
add  EAX EDX
pop  EBP
ret
```

Pre-coloured nodes

representing registers

nodes

- register = pre-coloured node
- no simplify, no spill
- coalesce possible

edges

- all registers interfere with each other
- explicit usage of registers
- call and return instructions influence liveness

V

summary

Summary
lessons learned

How can we assign registers to local variables and temporaries?

- perform liveness analysis
- build interference graph
- colour interference graph

What to do if the graph is not colourable?

- keep local variables in memory

How to handle move instructions efficiently?

- coalesce nodes safely

Literature

learn more

Andrew W. Appel, Jens Palsberg: Modern Compiler Implementation in Java, 2nd edition. 2002

Lal George, Andrew W. Appel: Iterative Register Coalescing. POPL 1996

Lal George, Andrew W. Appel: Iterative Register Coalescing. TOPLAS 18(3), 1996

Outlook
coming next

imperative and object-oriented languages

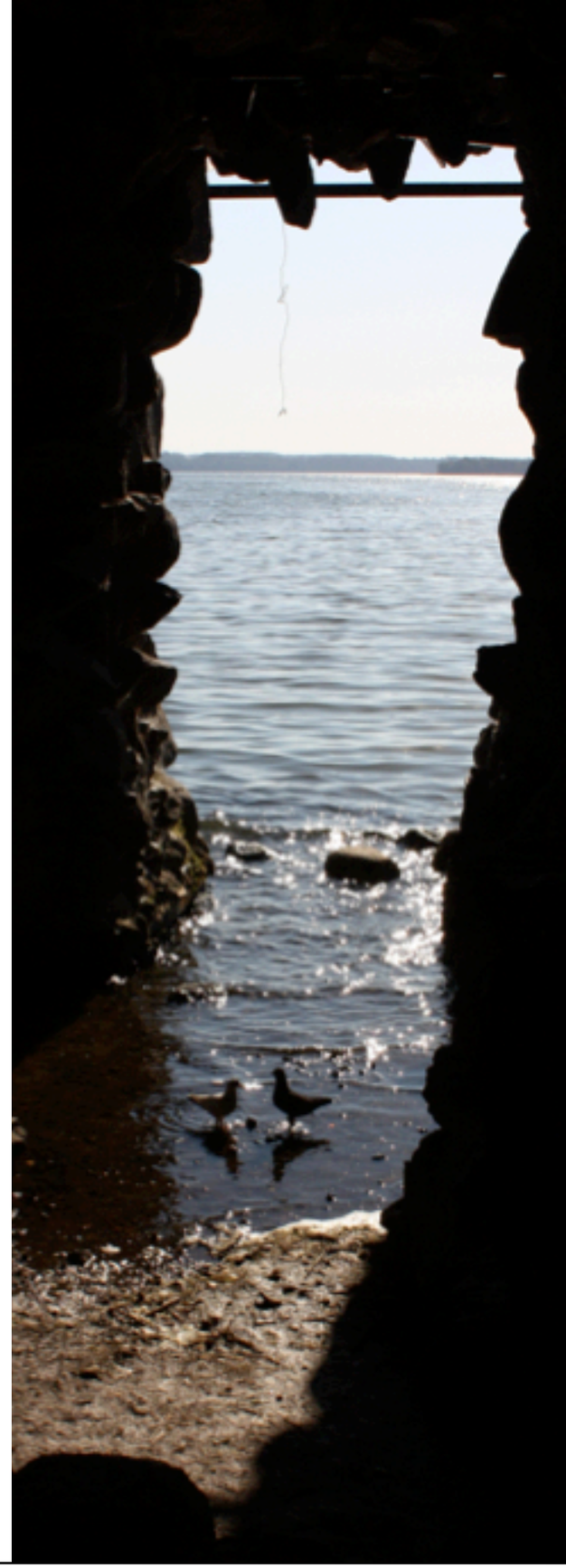
- Lecture 11: Register Allocation today
- Lecture 12: Garbage Collection Nov 13

compiler components & their generators

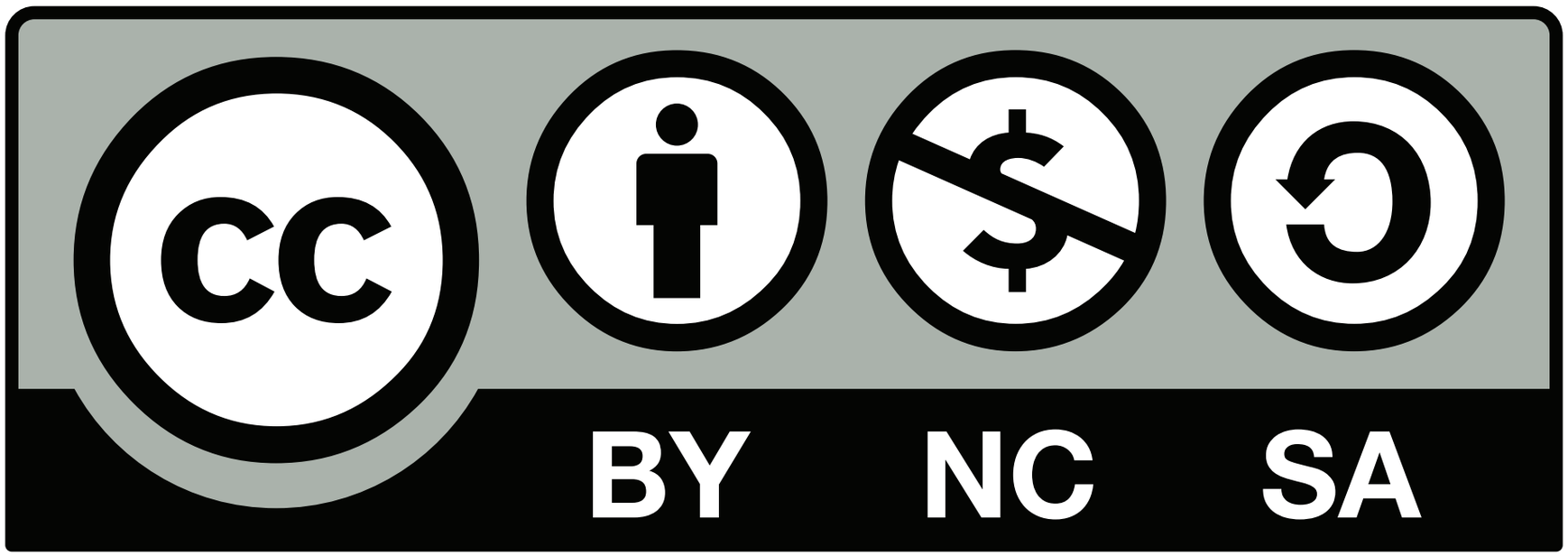
- Lecture 13: Scanning Nov 20
- Lecture 14: LL Parsing Dec 04
- Lecture 15: LR Parsing Dec 11

exam preparation

- Question & Answer Jan 08



copyrights


BY
NC
SA

Pictures

attribution & copyrights

Slide 1:
Colors #2 by Carmelo Speltino, some rights reserved

Slide 40:
Tchibo by Dominica Williamson, some rights reserved

Slide 76:
Romantic Pigeon Date by Harald Hoyer, some rights reserved