P_1V_{I1} = a1 P_1V_{I2} = a2

P_1

P_1V_{O1} = b1 P_1V_{O2} = b2

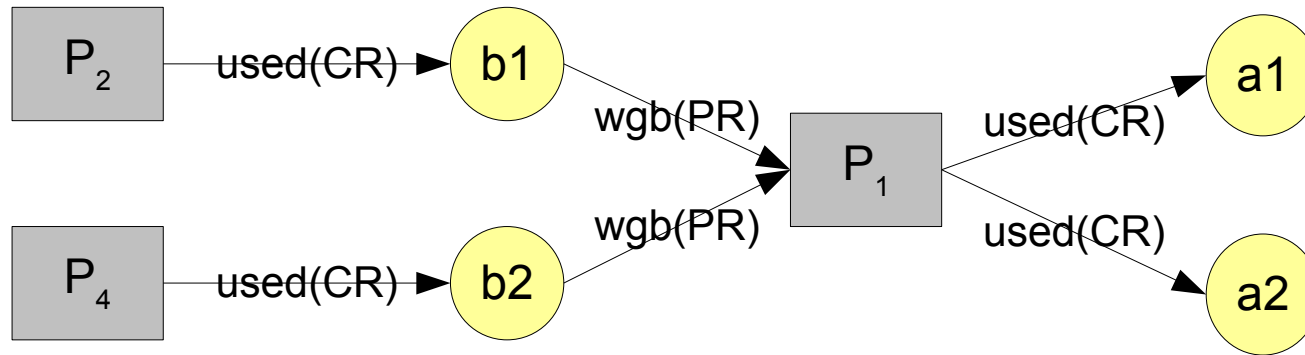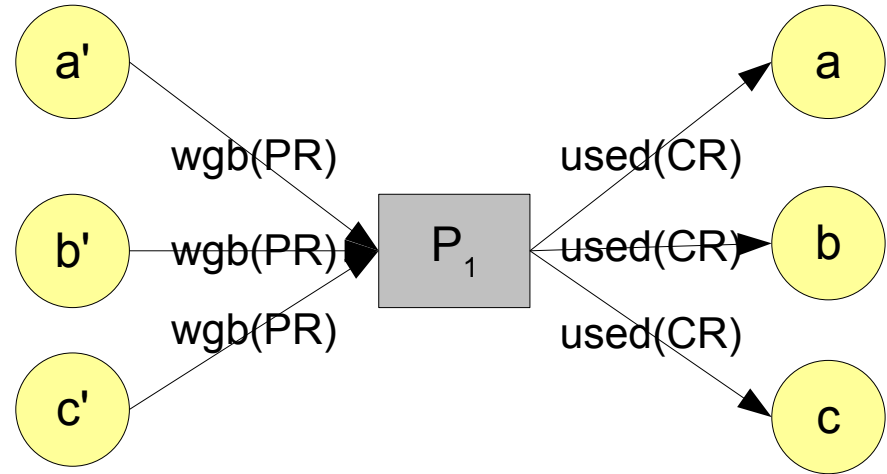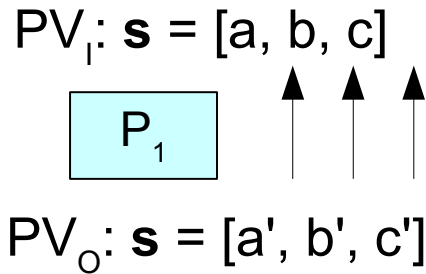P_2V_{I1} = b1 P_4V_{I1} = b2

P_2 P_4

- ok to represent specific lineage graphs
  - eg results of queries

-
- processor I/O var names also lost ("P_2V_{I1}")

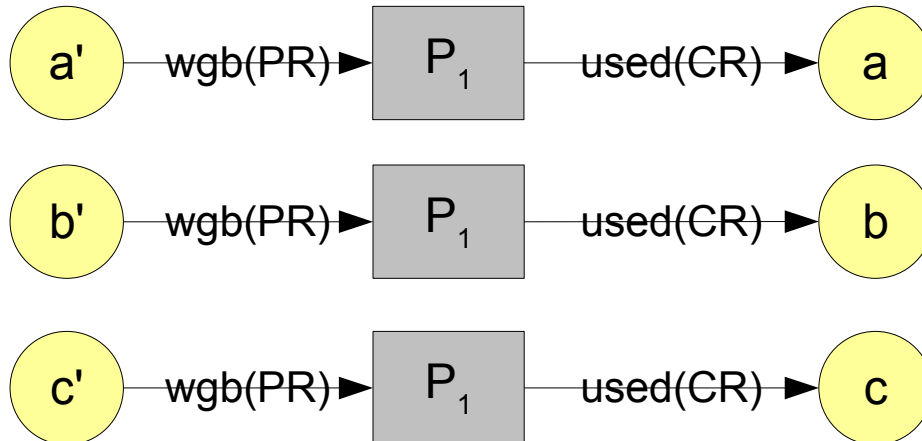- fine as long as granularity is constant
  - i.e. no collections

P_2 — used(CR) → b1 — wgb(PR) → P_1 — used(CR) → a1

P_4 — used(CR) → b2 — wgb(PR) → P_1 — used(CR) → a2

PR = "producer role"
CR = "consumer role"

PV$_I$: **s** = [a, b, c]

P$_1$

PV$_O$: **s** = [a', b', c']

a' — wgb(PR) → P$_1$ — used(CR) → a
b' — wgb(PR) → P$_1$ — used(CR) → b
c' — wgb(PR) → P$_1$ — used(CR) → c

this may lose information

is this valid OPM?:

a' — wgb(PR) → P$_1$ — used(CR) → a

b' — wgb(PR) → P$_1$ — used(CR) → b

c' — wgb(PR) → P$_1$ — used(CR) → c

and also:

L = [a,b,c]
L' = [a', b', c']

need to capture:
a $\in$ L, b $\in$ L, c $\in$ L
a' $\in$ L', b' $\in$ L', c' $\in$ L'

PV_I: **l(s)** = [a, b, c]

P

PV_O: **l(s)** = [x, y]

PV_O: l(s) = [a',b',c']

only useful annotation:
P is **index-preserving**:
$$PV_O[i] = PV_I[i]$$
$$\text{lineage}(PV_O[i]) = PV_I[i]$$

The annotation is used at query time to retain fine granularity

would it be useful to push it into the provenance graph?

how about using the annotation to deal with the granularity issue in the previous slide

[a,b,c] ─ wgb(PR) ▶ P_1 ─ used(CR) ▶ [a', b', c']

hasAnnotation

P_1 is
index-preserving