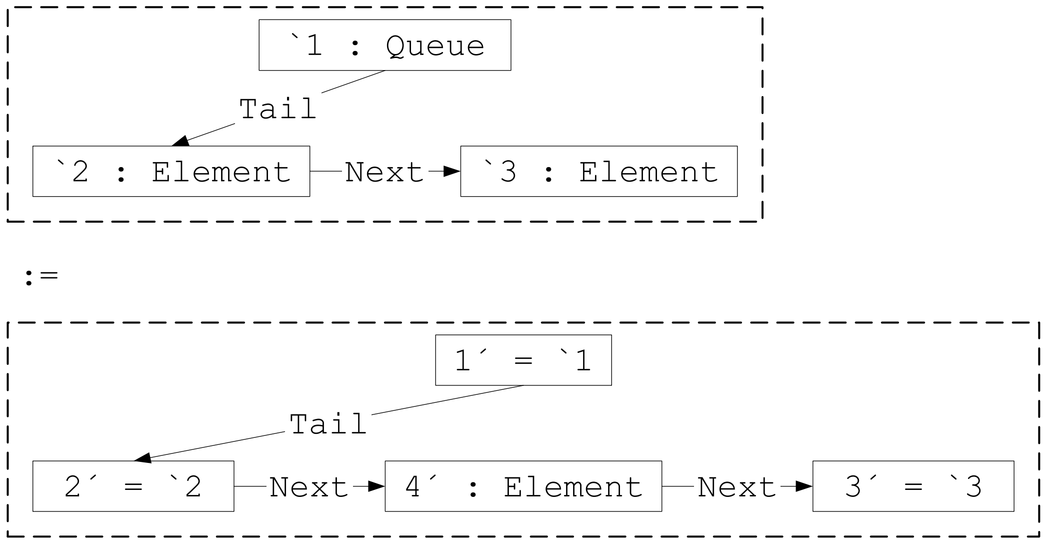
Solution
Sheet 5

Exercise 7

Graph rewrite rule `enqueue`

`production enqueue =`

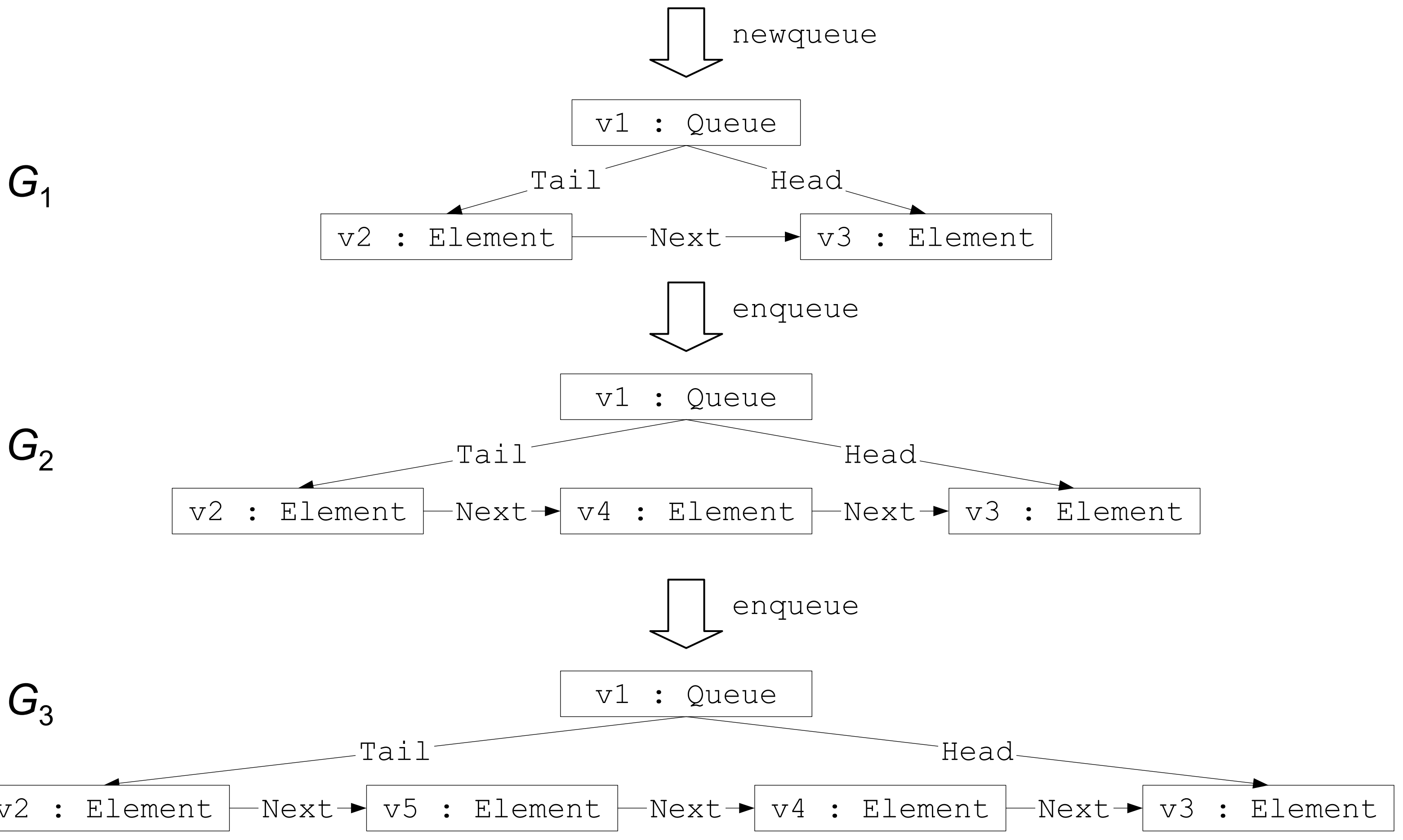


`end;`

Formal definition of `enqueue`

- $r = (L, K, R)$
- $L = (V_L, E_L, l_L)$
  - » $V_L = \{1, 2, 3\}$
  - » $E_L = \{(1, \text{Tail}, 2), (2, \text{Next}, 3)\}$
  - » $l_L = \{(1, \text{Queue}), (2, \text{Element}), (3, \text{Element})\}$
- $K = (V_K, E_K, l_K)$
  - » $V_K = \{1, 2, 3\}$
  - » $E_K = \{(1, \text{Tail}, 2)\}$
  - » $l_K = \{(1, \text{Queue}), (2, \text{Element}), (3, \text{Element})\}$
- $R = (V_R, E_R, l_R)$
  - » $V_R = \{1, 2, 3, 4\}$
  - » $E_R = \{(1, \text{Tail}, 2), (2, \text{Next}, 4), (4, \text{Next}, 3)\}$
  - » $l_R = \{(1, \text{Queue}), (2, \text{Element}), (3, \text{Element}), (4, \text{Element})\}$

Derivation


newqueue
G1
v1 : Queue
Tail
Head
v2 : Element
Next
v3 : Element
enqueue
G2
v1 : Queue
Tail
Head
v2 : Element
Next
v4 : Element
Next
v3 : Element
enqueue
G3
v1 : Queue
Tail
Head
v2 : Element
Next
v5 : Element
Next
v4 : Element
Next
v3 : Element

Formal definition of G_2

- $G_2 = (V_2, E_2, l_2)$
  - » V_2 = {v1, v2, v3, v4}
  - » E_2 = {(v1, Tail, v2), (v1, Head, v3), (v2, Next, v4), (v4, Next, v3)}
  - » l_2 = {(v1, Queue), (v2, Element), (v3, Element), (v4, Element)}

Application of `enqueue` to G_2

- Definition of an isomorphism $h : L \rightarrow G_L$, where G_L is a partial graph of G_2
  - » $G_L = (V_{GL}, E_{GL}, l_{GL})$
    - ⇨ V_{GL} = {v1, v2, v4}
    - ⇨ E_{GL} = {(v1, Tail, v2), (v2, Next, v4)}
    - ⇨ l_{GL} = {(v1, Queue), (v2, Element), (v4, Element)}
  - » $h : V_L \rightarrow V_{GL}$ = {(1, v1), (2, v2), (3, v4)}
- Intermediate graph
  $H = G_2 \setminus\setminus (h(L) \setminus h(K)) = G_2 \setminus\setminus (\emptyset$, {(v2, Next, v4)}, $\emptyset)$
  - » H is obtained from G_2 by deleting the edge (v2, Next, v4)
- Final graph
  $G_3 = H \oplus h'(R \setminus K)$ =
  $H \oplus h'$({4}, {(2, Next, 4), (4, Next,3)}, {(4, Element)}) =
  $H \oplus$ ({v5}, {(v2, Next, v5), (v5, Next, v4)}, {(v5, Element)})
  - » G_3 is obtained from H by adding the new node v5 and its incoming and outgoing Next edges

Other variants of graph rewrite rules

- Replacement of subgraphs rather than partial graphs
  - ⇨ Two versions of `enqueue` required
    - `enqueue1`: `enqueue` into empty queue (including `Head` edge)
    - `enqueue2`: `enqueue` into non-empty queue (without `Head` edge)
- Homomorphisms rather than isomorphisms
  - ⇨ Does not extend applicability of graph rewrite rules
    - Only `Element` nodes could be identified
    - In case of identification, a cycle of `Next` edges (of length 1 or 2) would have to be present in the host graph
    - There is no rule which creates a cycle