Grundlagen der Künstlichen Intelligenz

15. Handlungsplanung: Delete-Relaxierung

Malte Helmert

Universität Basel

6. Mai 2013

Woher kommen Heuristiken?

Eine einfache Planungsheuristik

STRIPS (Fikes & Nilsson, 1971) verwendete die Anzahl noch nicht erreichter Ziele in einer STRIPS-Planungsaufgabe als Heuristik:

$$h(s) := |G \setminus s|.$$

Intuition: weniger unerreichte Ziele $\rightsquigarrow$ näher an einem Zielzustand

$\rightsquigarrow$ STRIPS-Heuristik (Eigenschaften?)

Kritik an der STRIPS-Heuristik

Welche Probleme hat die STRIPS-Heuristik?

- recht uninformativ:
  wenn nicht sofort ein Ziel erreicht werden kann, haben alle Nachfolger denselben Heuristikwert.
- ignoriert fast die gesamte Problemstruktur:
  Der Heuristikwert hängt überhaupt nicht von den Aktionen ab!

$\rightsquigarrow$ wir brauchen einen bessere allgemeine Methode, um Heuristiken zu entwickeln

Handlungsplanung: Überblick

Kapitelüberblick:

- Einführung und Formalisierung $\rightsquigarrow$ Kapitel 14
- Drei Kernideen für allgemeine Heuristiken:
  - Delete-Relaxierung $\rightsquigarrow$ dieses Kapitel
  - Abstraktion $\rightsquigarrow$ Kapitel 16
  - Landmarken $\rightsquigarrow$ Kapitel 17

Handlungsplanung: Überblick

Kapitelüberblick:

- Einführung und Formalisierung $\rightsquigarrow$ Kapitel 14
- Drei Kernideen für allgemeine Heuristiken:
  - Delete-Relaxierung $\rightsquigarrow$ dieses Kapitel
  - Abstraktion $\rightsquigarrow$ Kapitel 16
  - Landmarken $\rightsquigarrow$ Kapitel 17

Grundidee Delete-Relaxierung

Schätze Lösungskosten durch Betrachten einer vereinfachten Planungsaufgabe ab, bei der unangenehme Aktionseffekte ignoriert werden.

Relaxierung für Planungsaufgaben

Relaxierte Planungsaufgaben: Idee

In STRIPS-Planungsaufgaben sind gute und schlechte Effekte leicht zu unterscheiden:

- Add-Effekte sind immer nützlich.
- Delete-Effekte sind immer schädlich.

Warum?

Relaxierte Planungsaufgaben: Idee

In STRIPS-Planungsaufgaben sind gute und schlechte Effekte leicht zu unterscheiden:

- Add-Effekte sind immer nützlich.
- Delete-Effekte sind immer schädlich.

Warum?

Idee für die Heuristik: Ignoriere alle Delete-Effekte.

Relaxierte Planungsaufgaben

Definition (Relaxierung von Aktionen)

Die Relaxierung a^+ einer STRIPS-Aktion a ist die Aktion mit $pre(a^+) = pre(a)$, $add(a^+) = add(a)$, $cost(a^+) = cost(a)$ und $del(a^+) = \emptyset$.

Definition (Relaxierung von Planungsaufgaben)

Die Relaxierung Π^+ einer STRIPS-Planungsaufgabe $\Pi = \langle V, I, G, A \rangle$ ist die Planungsaufgabe $\Pi^+ := \langle V, I, G, \{a^+ \mid a \in A\} \rangle$.

Definition (Relaxierung von Aktionsfolgen)

Die Relaxierung einer Aktionsfolge $\pi = a_1 \ldots a_n$ ist die Aktionsfolge $\pi^+ := a_1^+ \ldots a_n^+$.

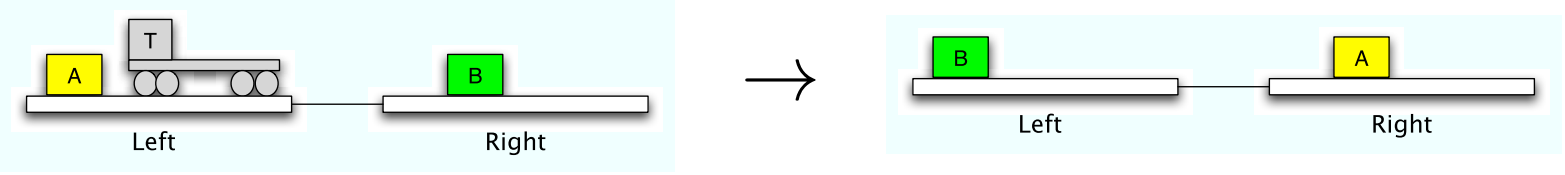
Relaxierte Planungsaufgaben: Terminologie

- STRIPS-Planungsaufgaben ohne Delete-Effekte heissen relaxierte Planungsaufgaben oder delete-freie Planungsaufgaben.
- Pläne für relaxierte Planungsaufgaben heissen relaxierte Pläne.
- Wenn Π eine STRIPS-Planungsaufgabe ist und π^+ ein Plan für Π^+ ist, dann heisst π^+ relaxierter Plan für Π.

Relaxierte Planungsaufgaben: Terminologie

- STRIPS-Planungsaufgaben ohne Delete-Effekte heissen relaxierte Planungsaufgaben oder delete-freie Planungsaufgaben.
- Pläne für relaxierte Planungsaufgaben heissen relaxierte Pläne.
- Wenn Π eine STRIPS-Planungsaufgabe ist und π^+ ein Plan für Π^+ ist, dann heisst π^+ relaxierter Plan für Π.
- Mit $h^+(\Pi)$ bezeichnen wir die Kosten eines optimalen Plans für Π^+, also eines optimalen relaxierten Plans.
- analog: $h^+(s)$ Kosten für optimalen relaxierten Plan für modifiziertes Problem mit Anfangszustand s statt I
- h^+ heisst optimale Relaxierungsheuristik.

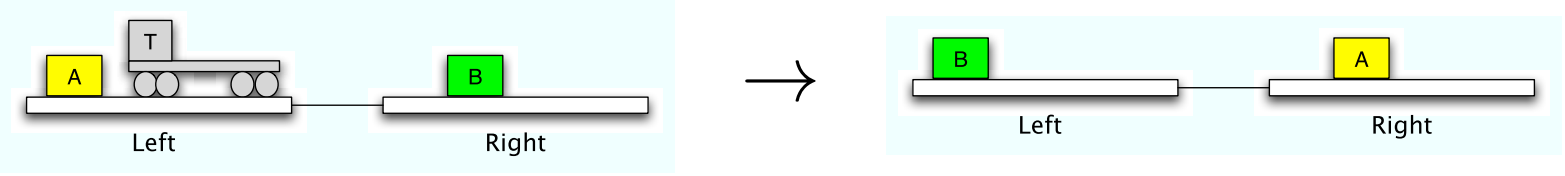
Beispiel: Logistik



Beispiel (Logistikproblem)

- Variablen: $V = \{at_{AL}, at_{AR}, at_{BL}, at_{BR}, at_{TL}, at_{TR}, in_{AT}, in_{BT}\}$
- Anfangszustand: $I = \{at_{AL}, at_{BR}, at_{TL}\}$
- Ziele: $G = \{at_{AR}, at_{BL}\}$
- Aktionen: $\{move_{LR}, move_{RL}, load_{AL}, load_{AR}, load_{BL}, load_{BR}, unload_{AL}, unload_{AR}, unload_{BL}, unload_{BR}\}$
- ...

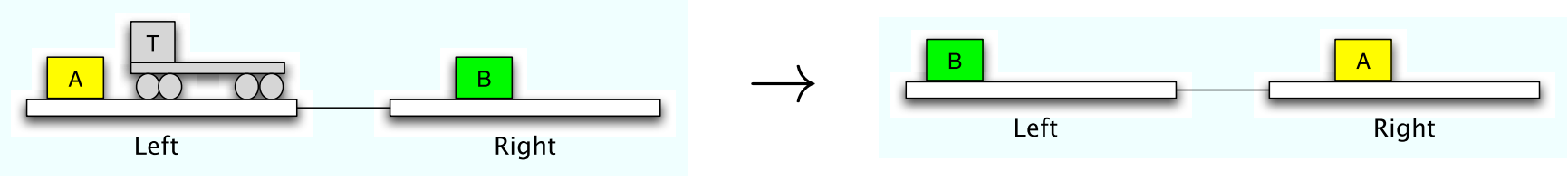
Beispiel: Logistik



Beispiel (Logistikproblem)

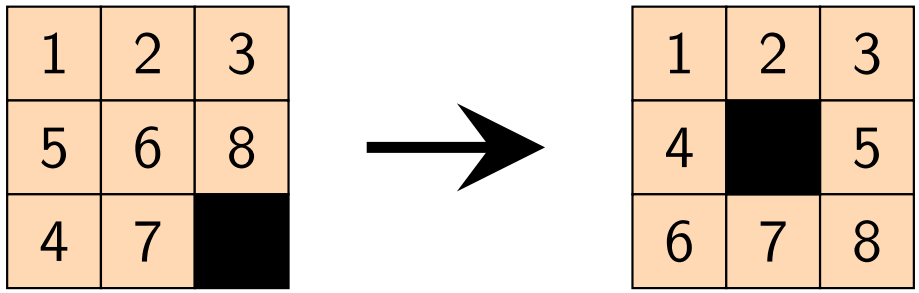
- $pre(move_{LR}) = \{at_{TL}\}$, $add(move_{LR}) = \{at_{TR}\}$, $del(move_{LR}) = \{at_{TL}\}$, $cost(move_{LR}) = 1$
- $pre(load_{AL}) = \{at_{TL}, at_{AL}\}$, $add(load_{AL}) = \{in_{AT}\}$, $del(load_{AL}) = \{at_{AL}\}$, $cost(load_{AL}) = 1$
- $pre(unload_{AL}) = \{at_{TL}, in_{AT}\}$, $add(unload_{AL}) = \{at_{AL}\}$, $del(unload_{AL}) = \{in_{AT}\}$, $cost(unload_{AL}) = 1$
- ...

Beispiel: Logistik



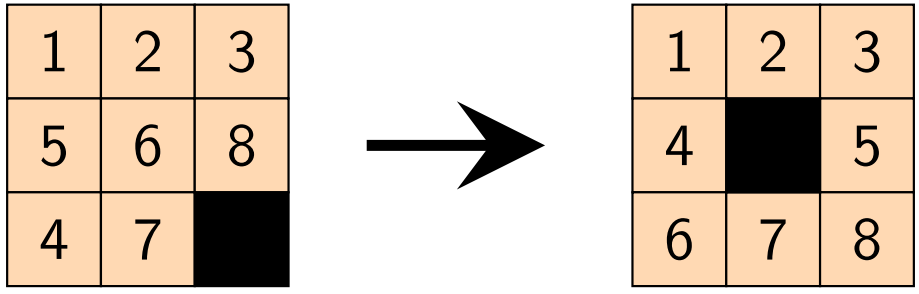
- optimaler Plan:
  1. $load_{AL}$
  2. $move_{LR}$
  3. $unload_{AR}$
  4. $load_{BR}$
  5. $move_{RL}$
  6. $unload_{BL}$
- optimaler relaxierter Plan: ?
- $h^*(I) = 6$, $h^+(I) = ?$

Beispiel: 8-Puzzle



- (Unvereinfachtes) Problem:
  - Eine Kachel kann von Feld A in Feld B bewegt werden, wenn A und B angrenzend sind und B frei ist.
- Vereinfachung, die der Manhattan-Distanz zugrunde liegt:
  - Eine Kachel kann von Feld A in Feld B bewegt werden, wenn A und B angrenzend sind.
- relaxiertes Problem:
  - Eine Kachel kann von Feld A in Feld B bewegt werden, wenn A und B angrenzend sind und B frei ist.
  - . . . wobei Delete-Effekte ignoriert werden (insbesondere: Felder, die jemals frei waren, bleiben frei)

Beispiel: 8-Puzzle



- tatsächlicher Zielabstand: $h^*(s) = 8$
- Manhattan-Distanz: $h^{\mathrm{MD}}(s) = 6$
- optimale Delete-Relaxierung: $h^+(s) = 7$

allgemein:
h^+ dominiert Manhattan-Distanz für das Schiebe-Puzzle
(d. h. $h^{\mathrm{MD}}(s) \leq h^+(s) \leq h^*(s)$ für alle Zustände s)

Relaxierte Lösungen: suboptimal oder optimal?

- Für allgemeine STRIPS-Planungsaufgaben ist h^+ eine zulässige und konsistente Heuristik.

Relaxierte Lösungen: suboptimal oder optimal?

- Für allgemeine STRIPS-Planungsaufgaben ist h^+ eine zulässige und konsistente Heuristik.
- Wie berechnen wir h^+ effizient?
  - Es ist einfach, delete-freie Planungsaufgaben suboptimal zu lösen. (Wie?)
  - Optimale Lösung (und damit Berechnung von h^+) ist aber NP-hart (einfache Reduktion von Set Cover).
- Praktische Heuristiken nähern h^+ von unten oder oben an.

Konkrete Relaxierungsheuristiken

Relaxierte Planungsgraphen

- relaxierte Planungsgraphen: repräsentieren,
  welche Variablen in Π^+ wie erreicht werden können
- Graphen mit Variablenebenen V^i und Aktionsebenen A^i
  - Variablenebene V^0 enthält Knoten v^0 für jedes $v \in I$
  - Aktionsebene A^{i+1} enthält Knoten a^{i+1} für Aktion a,
    wenn V^i den Knoten v^i für alle $v \in \mathit{pre}(a)$ enthält
  - Variablenebene V^{i+1} enthält Knoten v^{i+1}
    wenn vorige Variablenebene v^i enthält
    oder vorige Aktionsebene ein a^{i+1} mit $v \in \mathit{add}(a)$ enthält

Relaxierte Planungsgraphen (Fortsetzung)

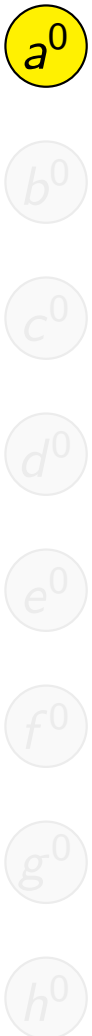
- Zielknoten G^i, wenn $v^i \in V^i$ für alle $v \in G$
- Graph kann für beliebig viele Ebenen konstruiert werden, wird aber nach endlich vielen Ebenen stabil
  $\rightsquigarrow V^{i+1} = V^i$ und $A^{i+1} = A^i$ (Warum?)
- gerichtete Kanten:
  - von v^i zu a^{i+1} wenn $v \in pre(a)$ (Vorbedingungskanten)
  - von a^i zu v^i, wenn $v \in add(a)$ (Effektkanten)
  - von v^i zu G^i, wenn $v \in G$ (Zielkanten)
  - von v^i zu v^{i+1} (No-Op-Kanten)

Illustrierendes Beispiel

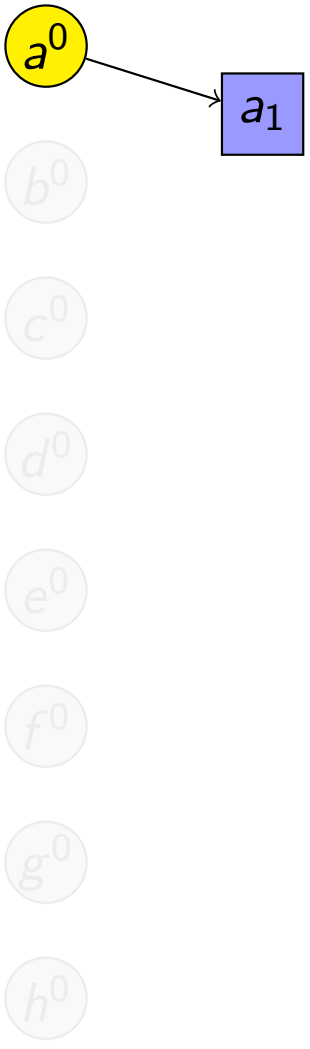
Schreibe im Folgenden Aktionen a mit $pre(a) = \{p_1, \ldots, p_k\}$, $add(a) = \{a_1, \ldots, a_l\}$, $del(a) = \emptyset$ und $cost(a) = c$ als $\langle p_1, \ldots, p_k \rightarrow a_1, \ldots, a_l \rangle_c$

$$V = \{a, b, c, d, e, f, g, h\}$$
$$I = \{a\}$$
$$G = \{c, d, e, f, g\}$$
$$A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$$
$$a_1 = \langle a \rightarrow b, c \rangle_3$$
$$a_2 = \langle a, c \rightarrow d \rangle_1$$
$$a_3 = \langle b, c \rightarrow e \rangle_1$$
$$a_4 = \langle b \rightarrow f \rangle_1$$
$$a_5 = \langle d \rightarrow e, f \rangle_1$$
$$a_6 = \langle d \rightarrow g \rangle_1$$

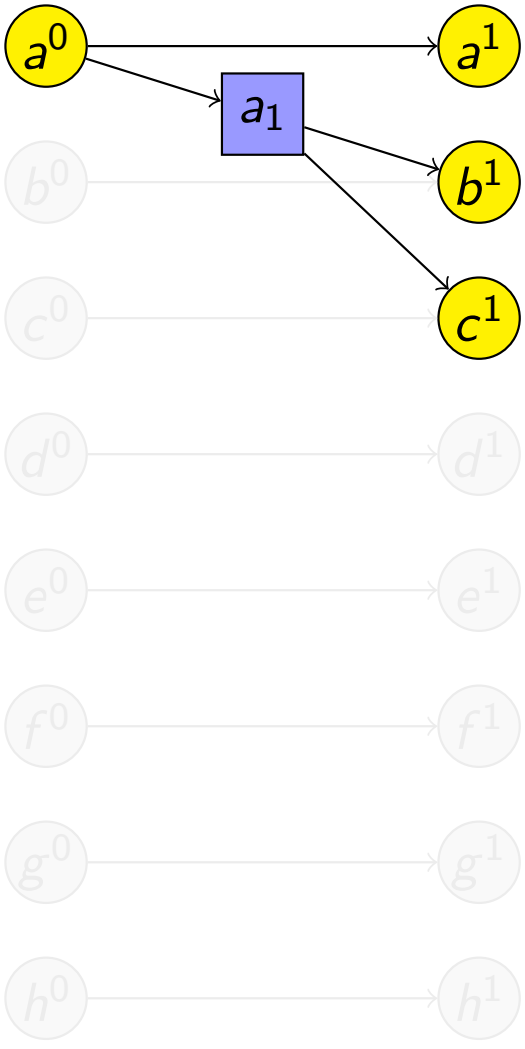
Illustrierendes Beispiel: relaxierter Planungsgraph



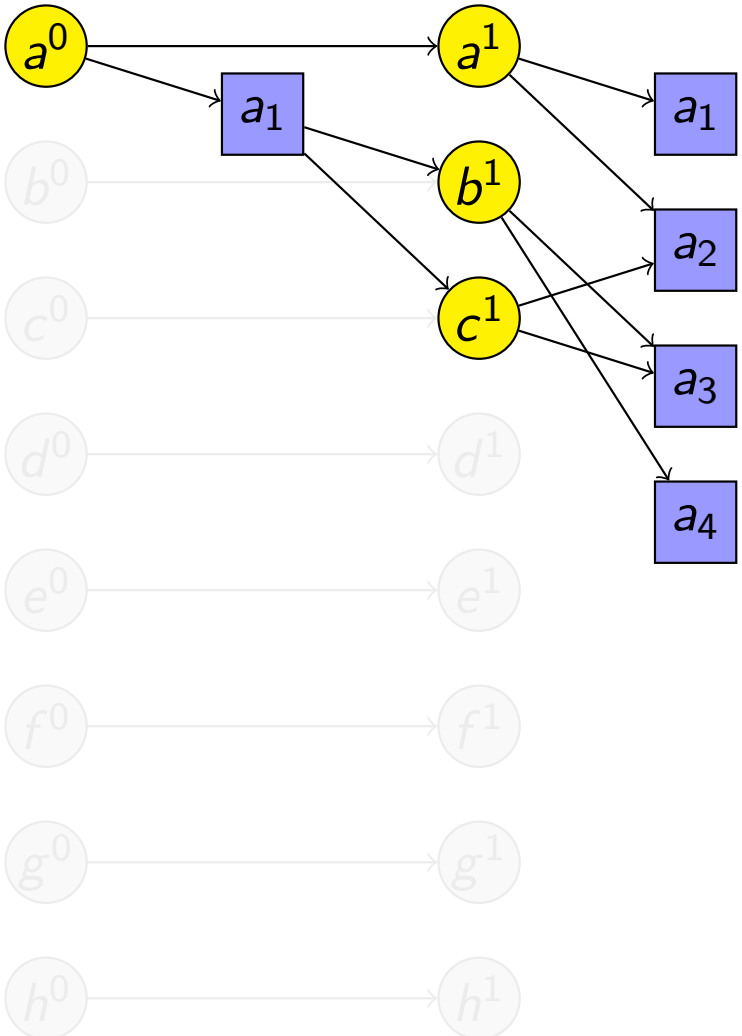
Illustrierendes Beispiel: relaxierter Planungsgraph



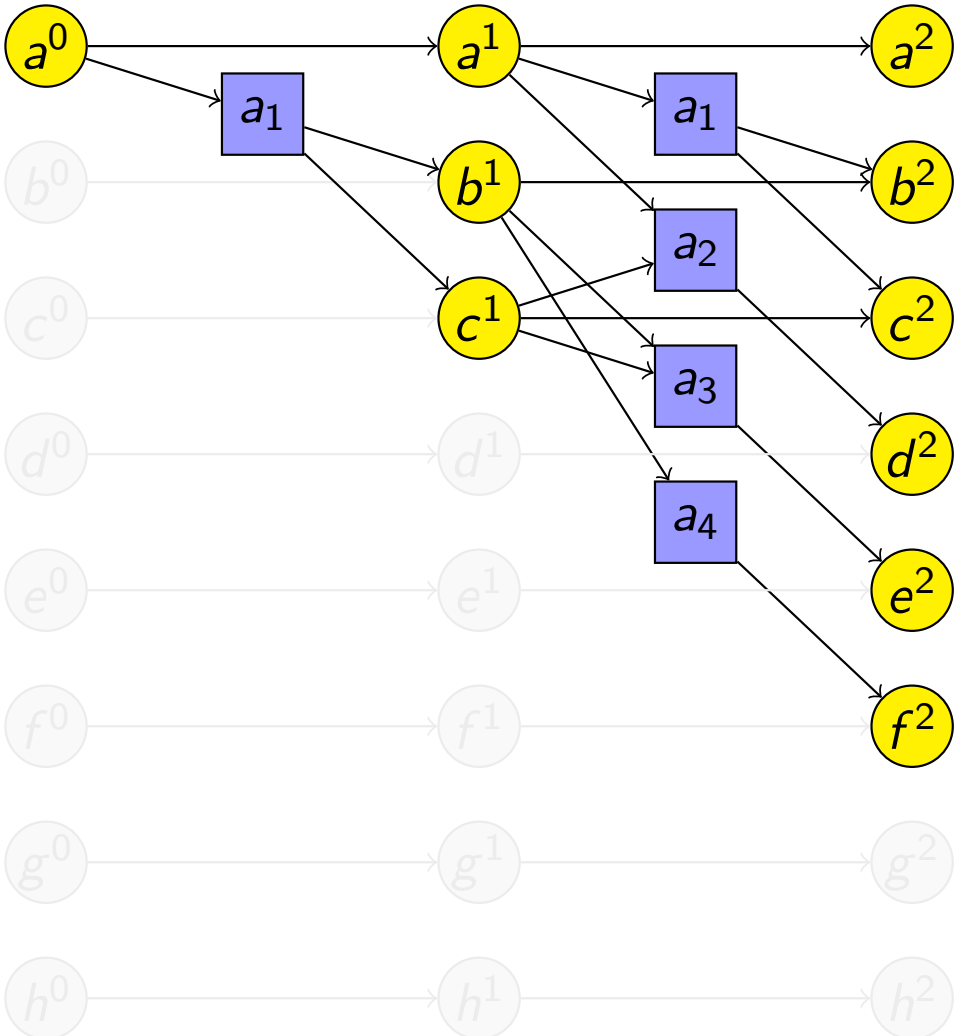
Illustrierendes Beispiel: relaxierter Planungsgraph



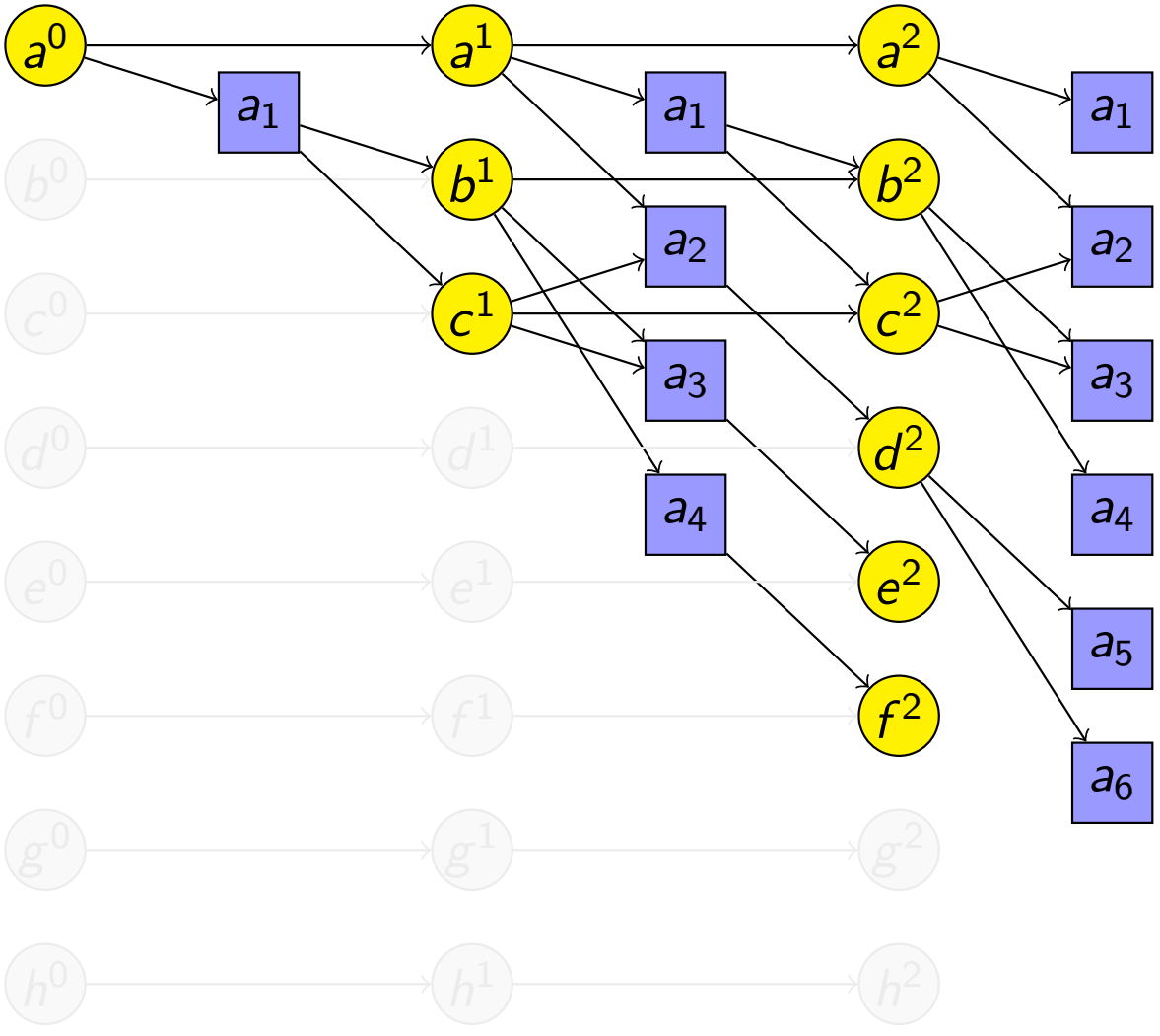
Illustrierendes Beispiel: relaxierter Planungsgraph



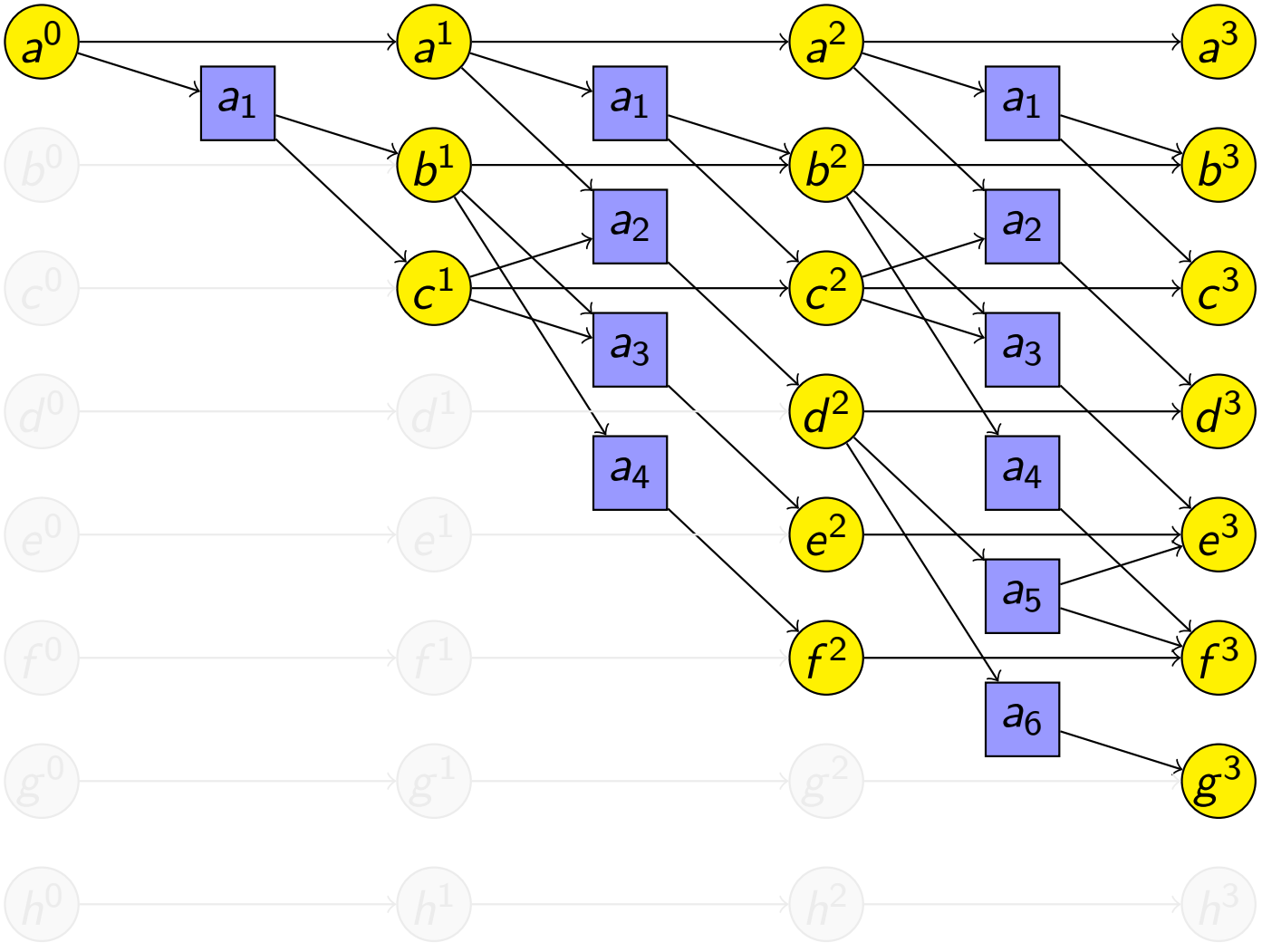
Illustrierendes Beispiel: relaxierter Planungsgraph



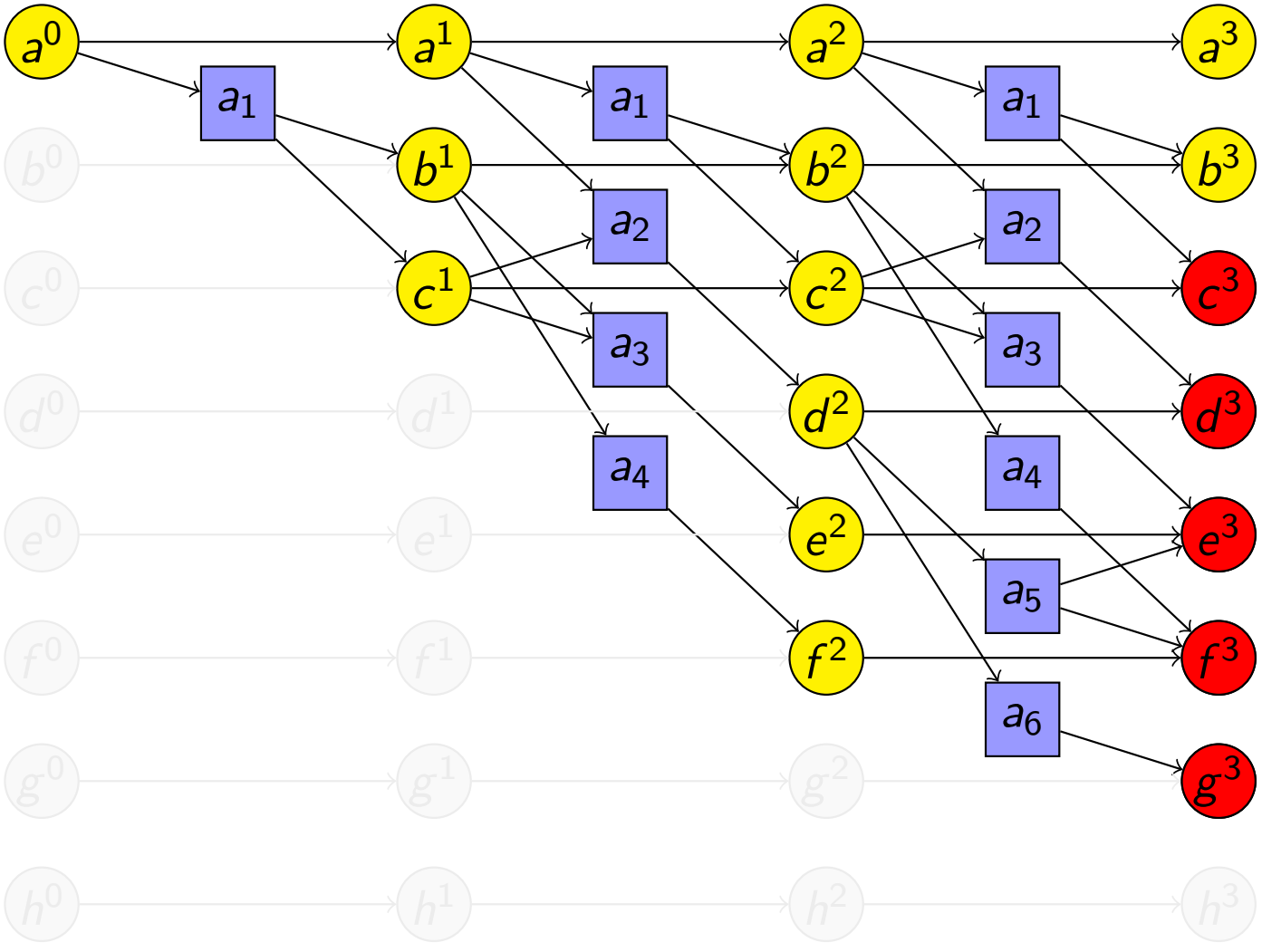
Illustrierendes Beispiel: relaxierter Planungsgraph



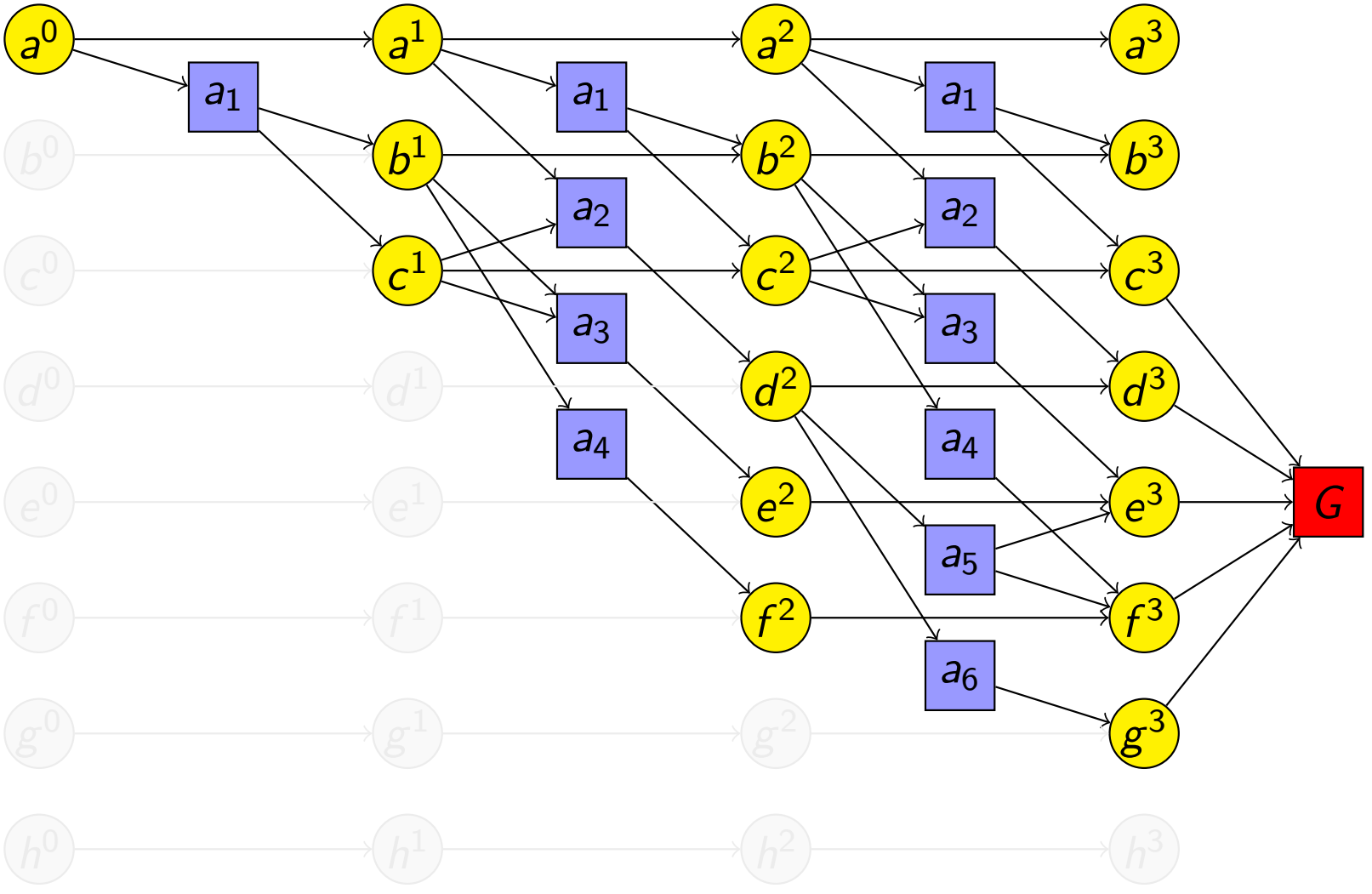
Illustrierendes Beispiel: relaxierter Planungsgraph



Illustrierendes Beispiel: relaxierter Planungsgraph



Illustrierendes Beispiel: relaxierter Planungsgraph



Generische Relaxierter-Planungsgraph-Heuristik

Heuristikwerte aus relaxierten Planungsgraphen

function *generic-rpg-heuristic*$(\langle V, I, G, A\rangle, s)$:
    $\Pi^+ := \langle V, s, G, A^+\rangle$
    for $k \in \{0, 1, 2, \dots\}$:
        $rpg := RPG_k(\Pi^+)$ [relaxed planning graph to layer k]
        if *rpg* contains a goal node:
            Annotate nodes of *rpg*.
            if termination criterion is true:
                return heuristic value from annotations
        else if graph has stabilized:
            return ∞

- $\rightsquigarrow$ allgemeines Muster für RPG-Heuristiken
- $\rightsquigarrow$ für konkrete Heuristik: hervorgehobene Stellen ausfüllen

Konkrete Beispiele für die generische RPG-Heuristik

Viele Planungsheuristiken passen in dieses allgemeine Muster.

In dieser Vorlesung:

- Maximums-Heuristik $h^{\max}$ (Bonet & Geffner, 1999)
- additive Heuristik h^{add} (Bonet, Loerincs & Geffner, 1997)
- FF-Heuristik h^{FF} (Hoffmann & Nebel, 2001)
  in der Variante von Keyder & Geffner (2008)

Anmerkung:

- Die effizientesten Implementierungen dieser Heuristiken verwenden keine relaxierten Planungsgraphen, sondern alternative (äquivalente) Definitionen.

Maximums- und additive Heuristik

- $h^{\max}$ und h^{add} sind die einfachsten RPG-Heuristiken
- Knotenannotationen sind Kostenwerte (= Zahlen)
- Die Kosten eines Knoten schätzen, wie teuer es ist,
  - eine gegebene Variable wahr zu machen
  - eine gegebene Aktion zu erreichen und auszuführen
  - das Ziel zu erreichen

Maximums- und additive Heuristik: ausgefülltes Muster

$h^{\max}$ und h^{add}

Berechnung der Annotationen:

- Kosten von Variablenknoten:
  0 in Ebene 0; sonst Minimum der Kosten der Vorgängerknoten
- Kosten von Aktions- und Zielknoten:
  Maximum ($h^{\max}$) bzw. Summe (h^{add})
  der Kosten der Vorgängerknoten;
  bei Aktionsknoten a^i addiere zusätzlich $cost(a)$

Terminierungskriterium:

- Stabilität: terminiere, wenn $V^i = V^{i-1}$ und Kosten aller Knoten in V^i gleich Kosten entsprechender Knoten in V^{i-1}

Heuristikwert:

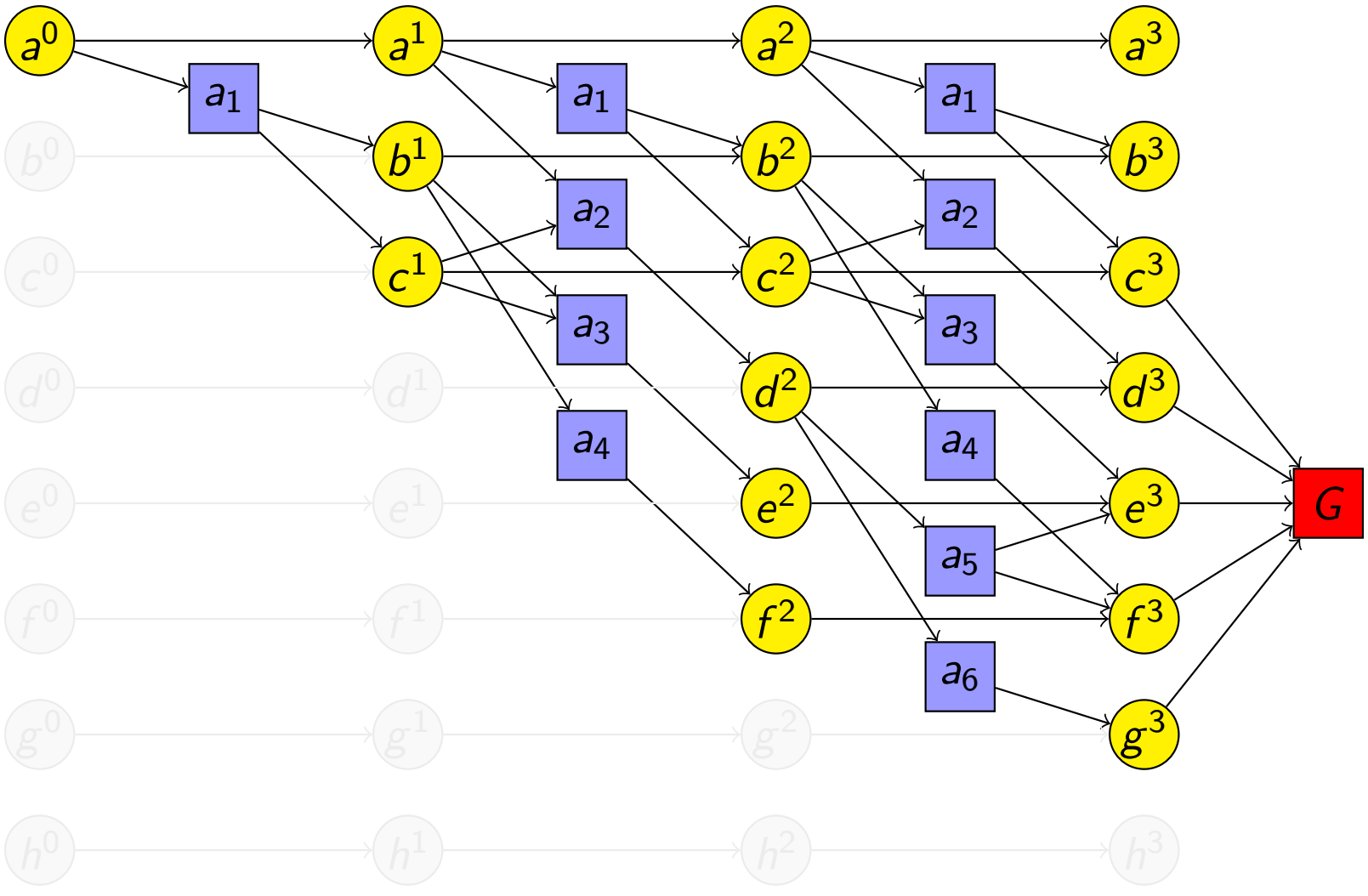
- der Wert des Zielknotens in der letzten Ebene

Maximums- und additive Heuristik: Intuition

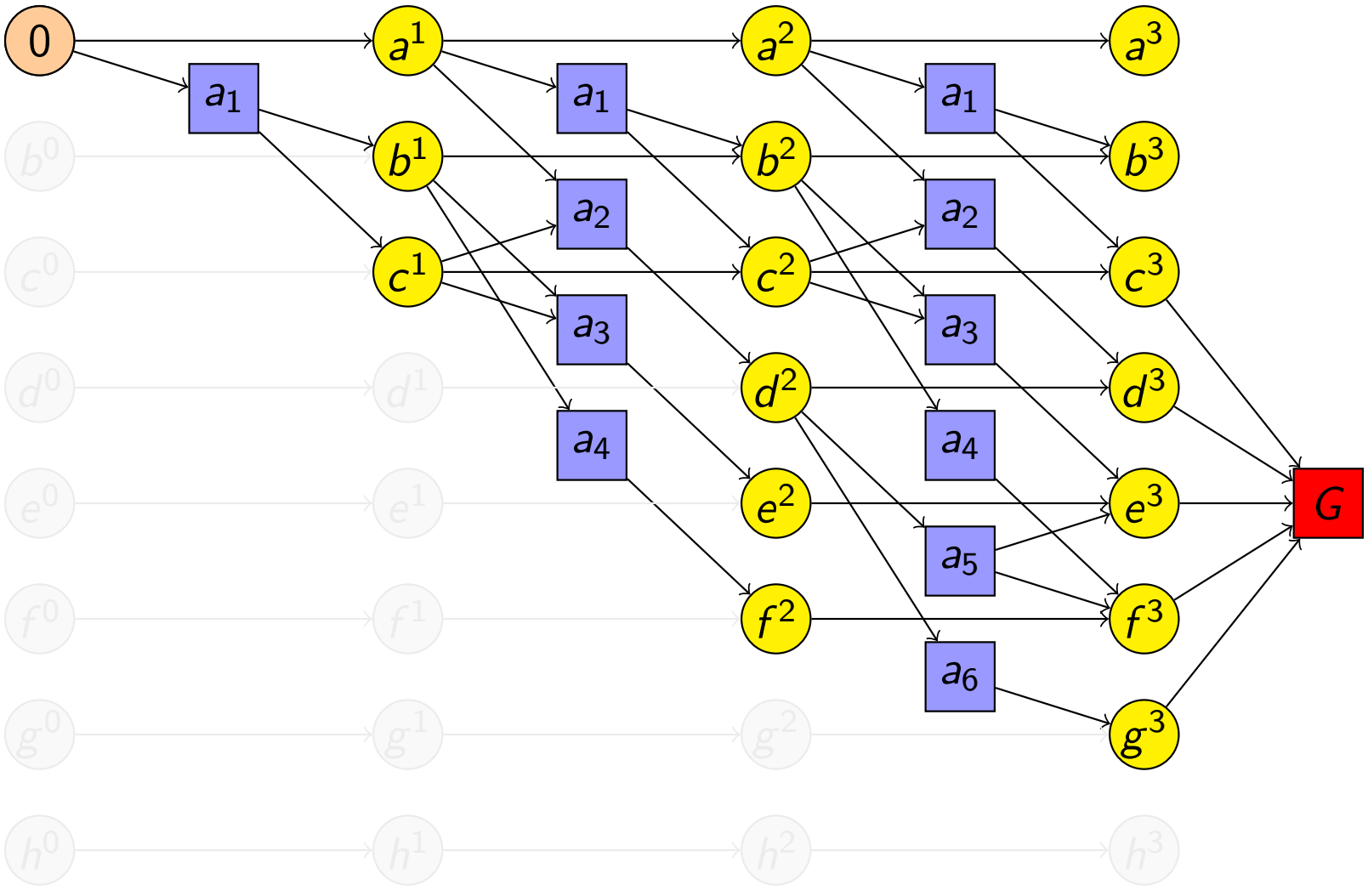
Intuition:

- Variablenknoten:
  - wähle billigste Möglichkeit, die Variable zu erreichen
- Aktionsknoten/Zielknoten:
  - $h^{\max}$ ist optimistisch:
    wenn wir die teuerste Vorbedingung erreichen,
    können wir nebenbei die anderen Vorbedingungen erreichen
    (daher Maximierung über Kosten)
  - h^{add} ist pessimistisch:
    alle Bedingungen müssen komplett unabhängig voneinander
    erreicht werden (daher Summierung über Kosten)

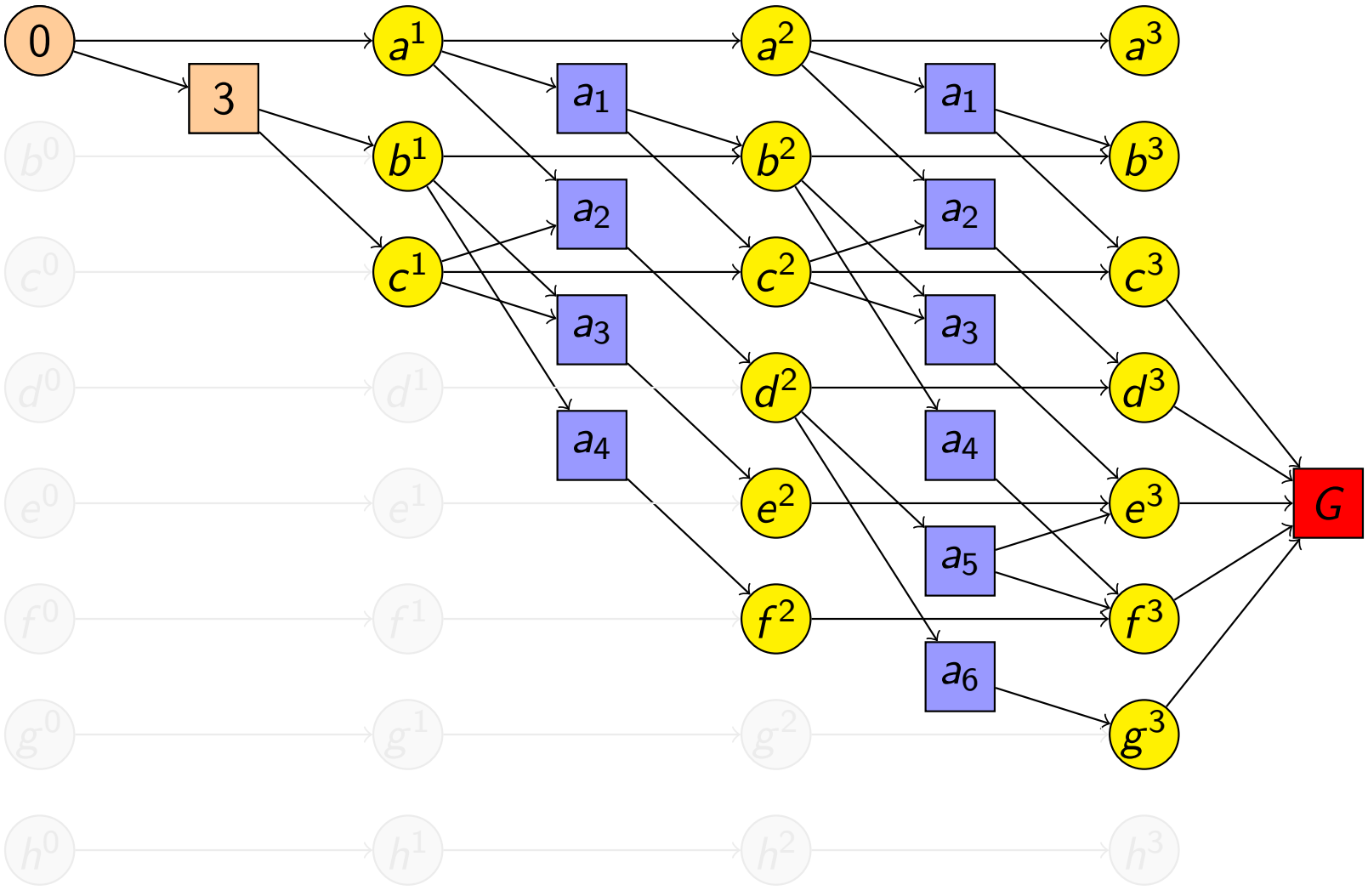
Illustrierendes Beispiel: $h^{\max}$



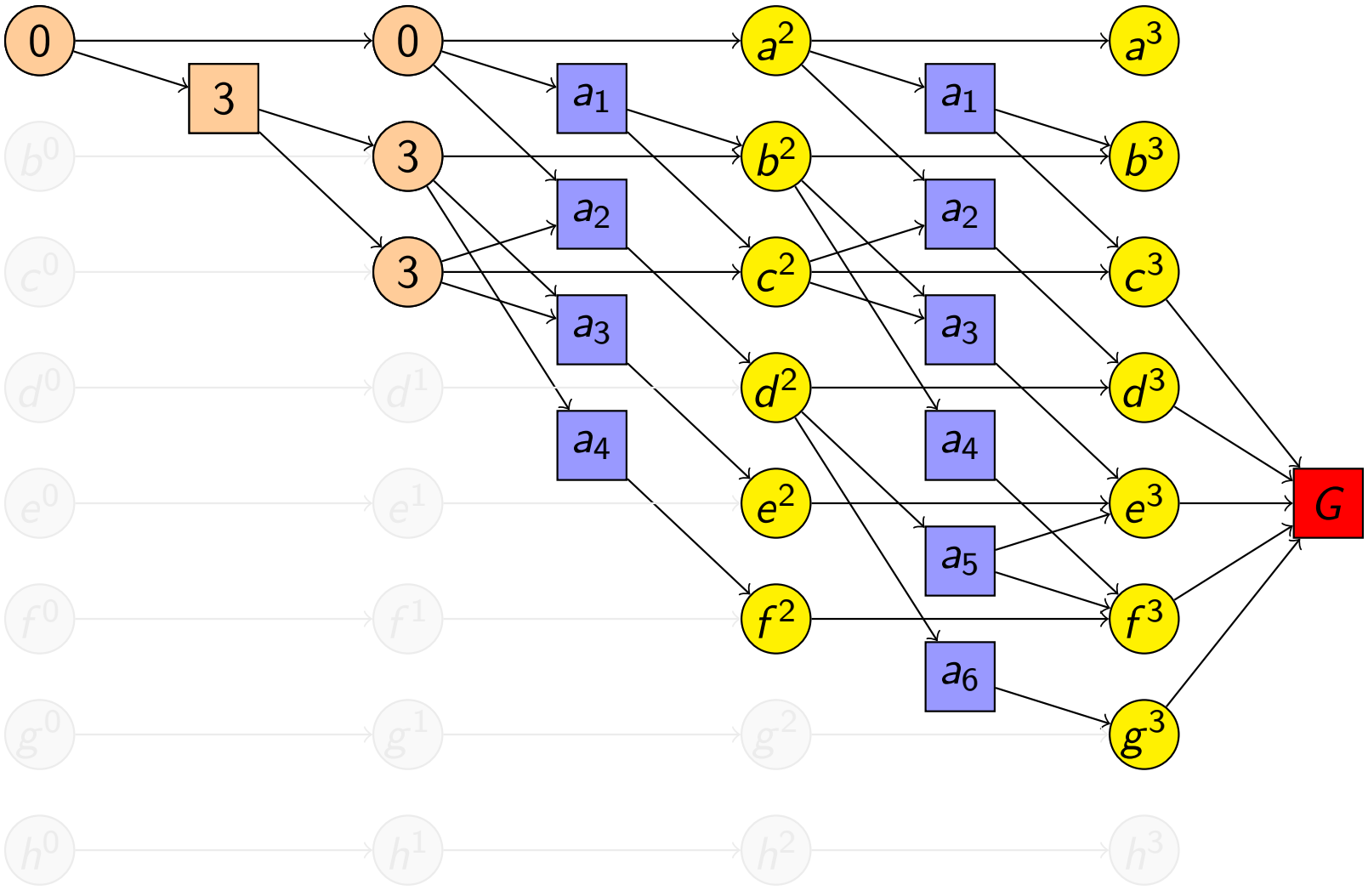
Illustrierendes Beispiel: $h^{\max}$



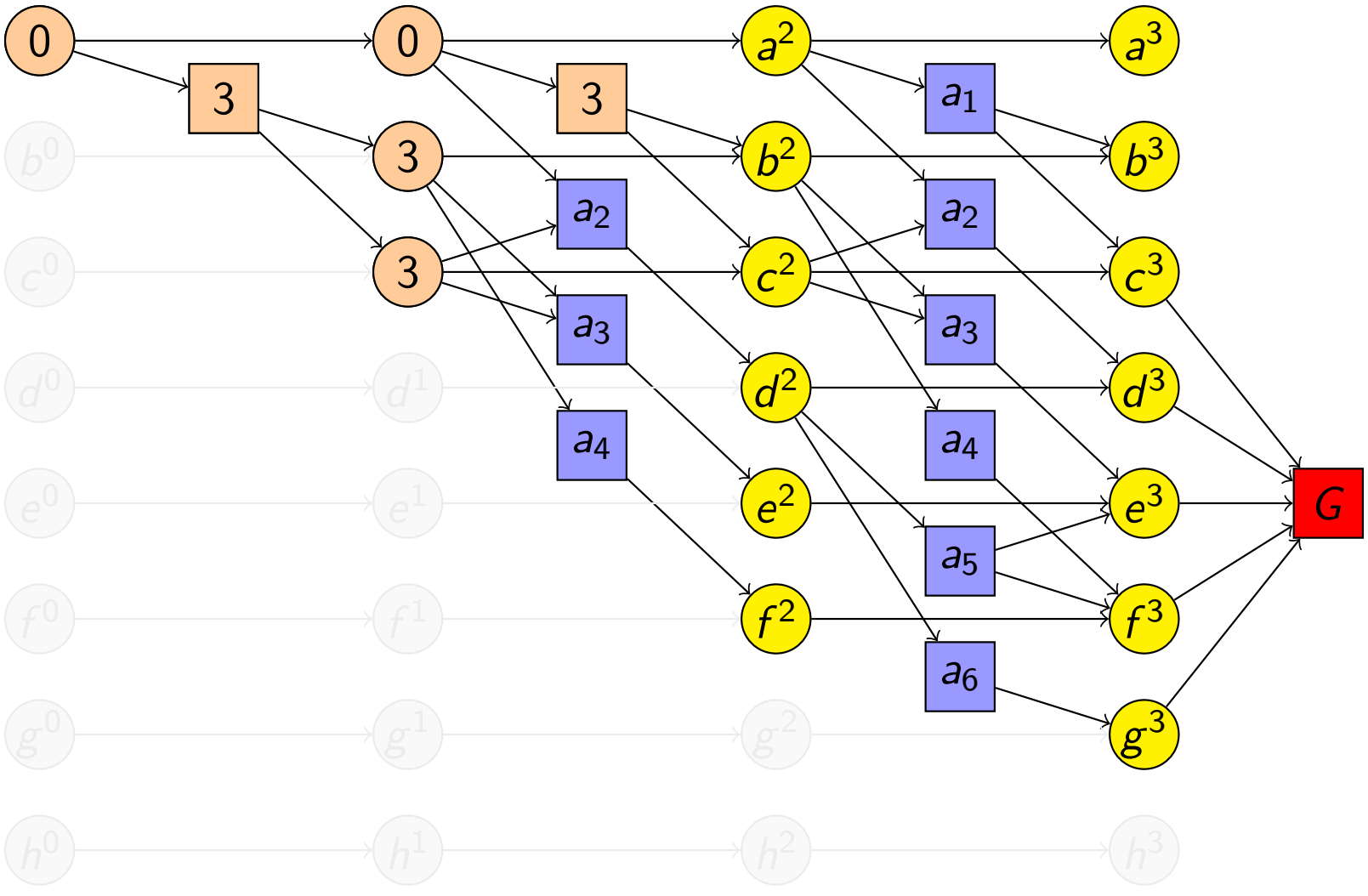
Illustrierendes Beispiel: $h^{\max}$



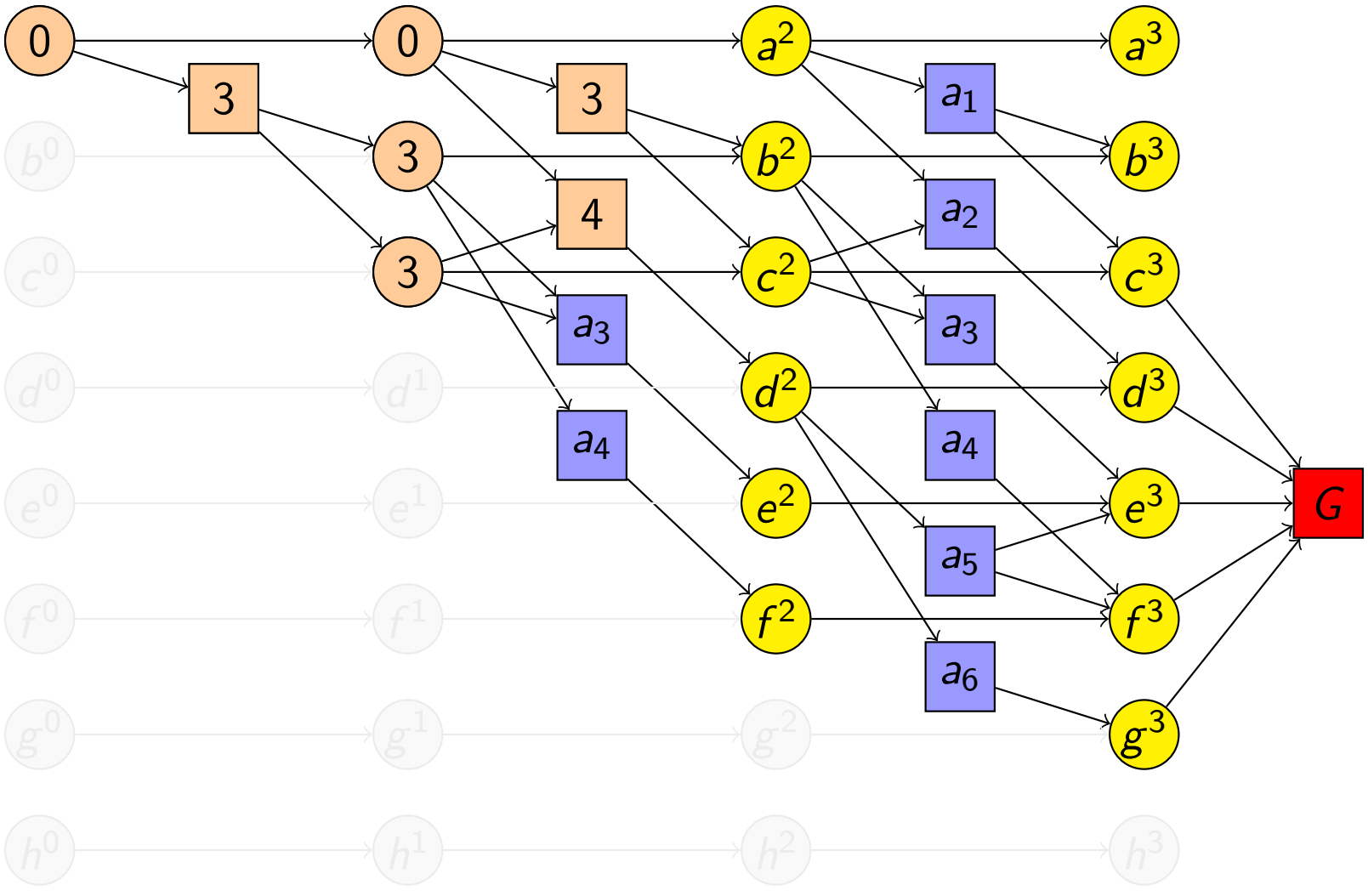
Illustrierendes Beispiel: $h^{\max}$



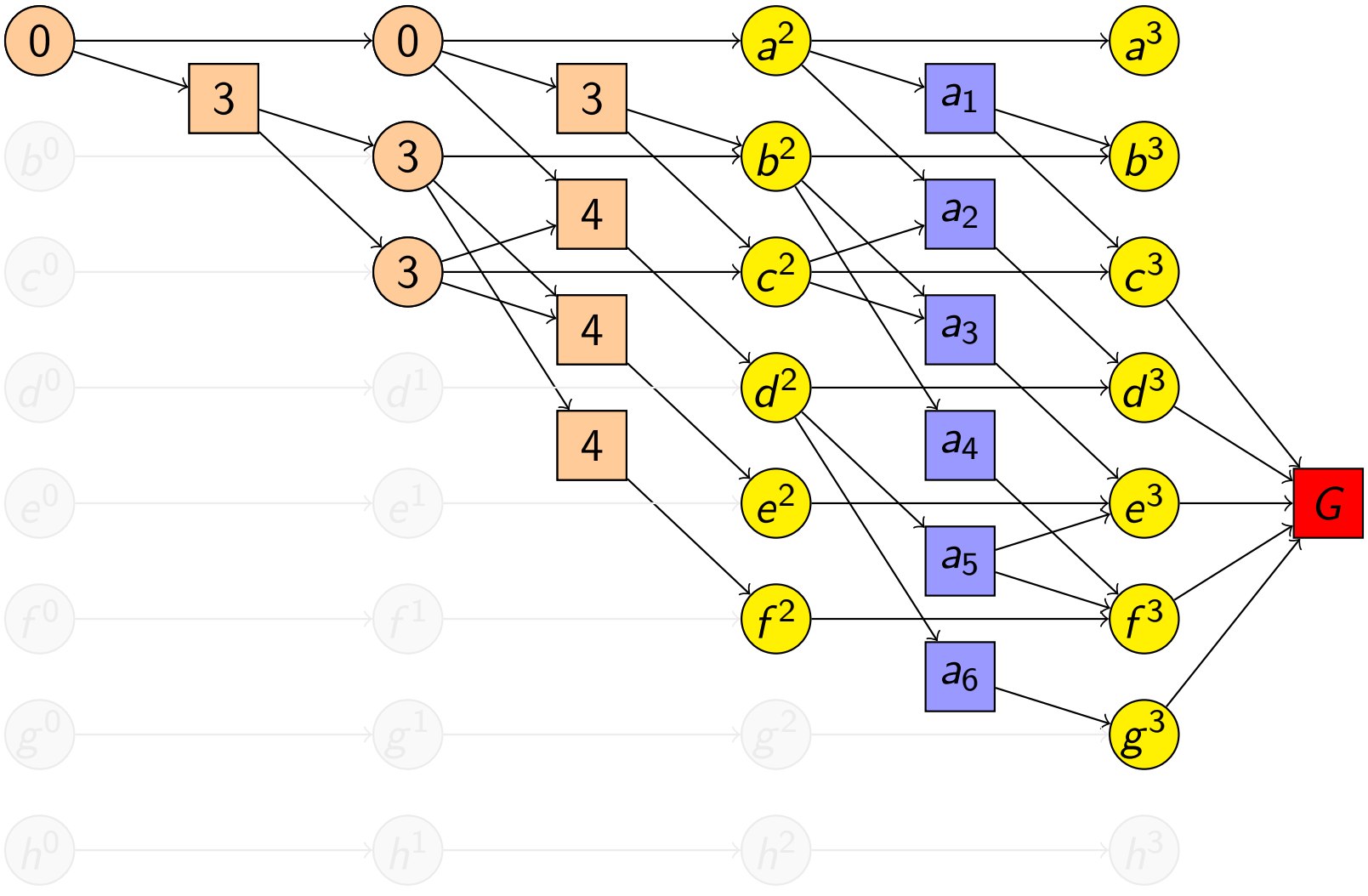
Illustrierendes Beispiel: $h^{\max}$



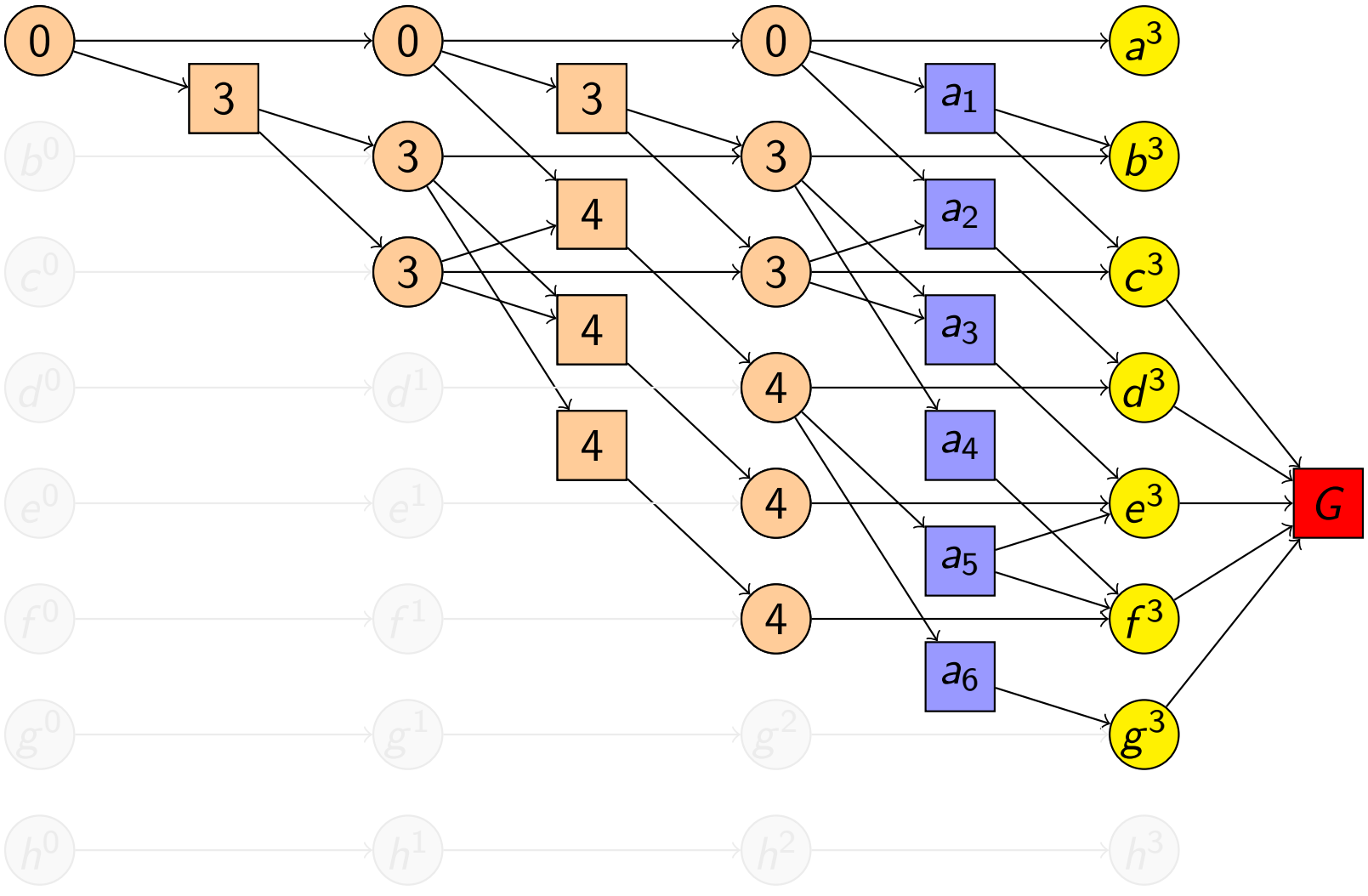
Illustrierendes Beispiel: $h^{\max}$



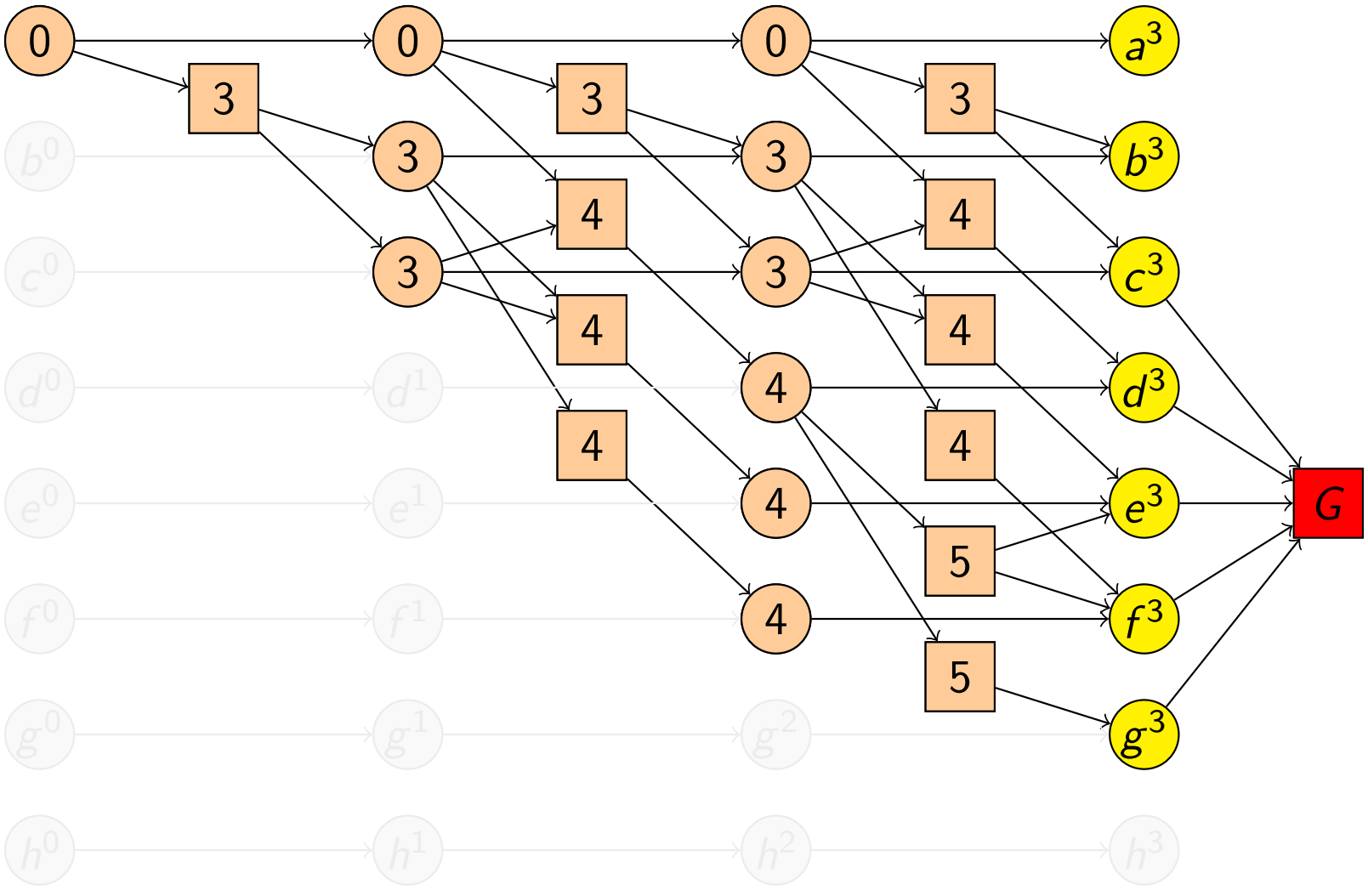
Illustrierendes Beispiel: $h^{\max}$



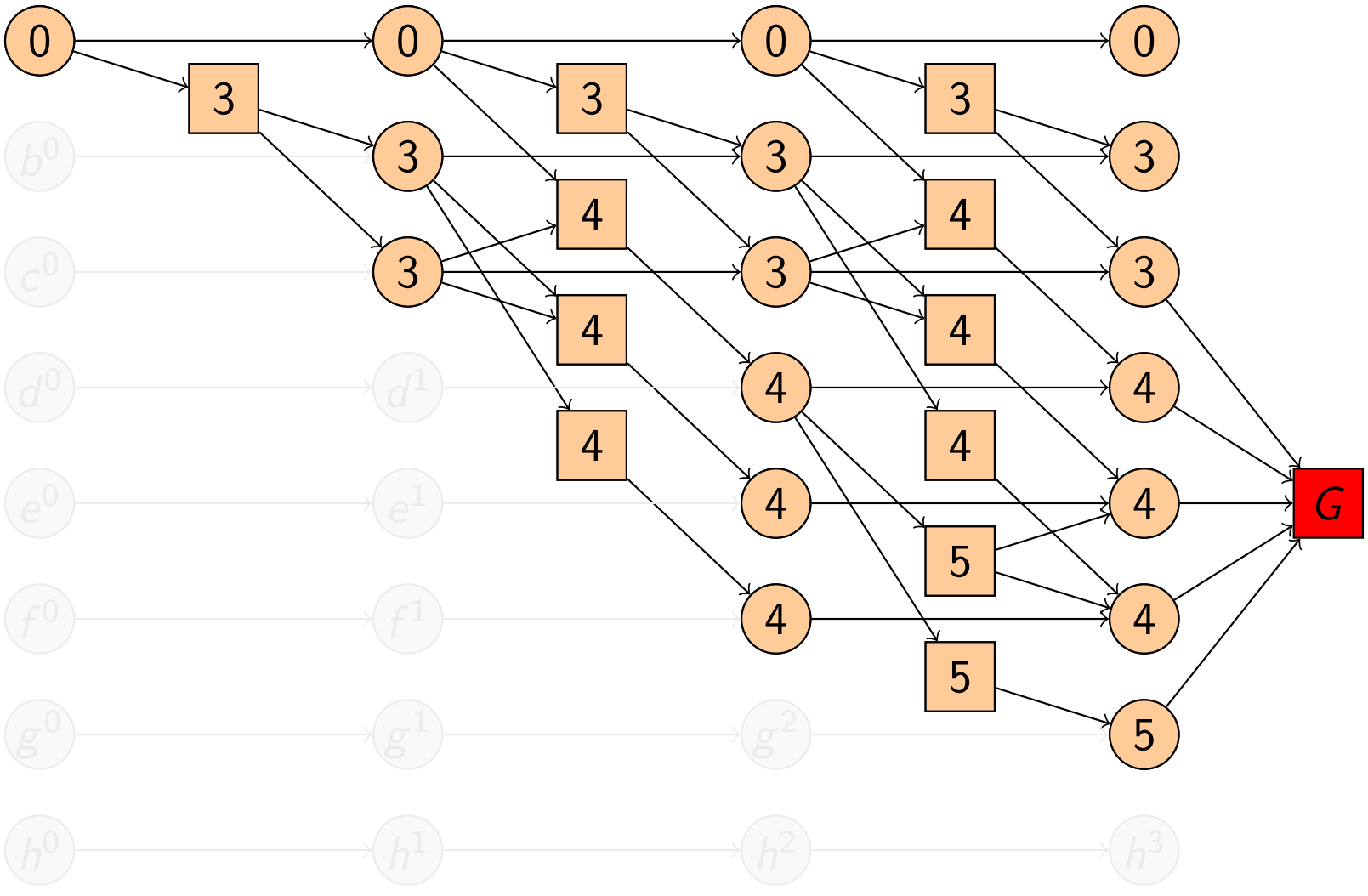
Illustrierendes Beispiel: $h^{\max}$



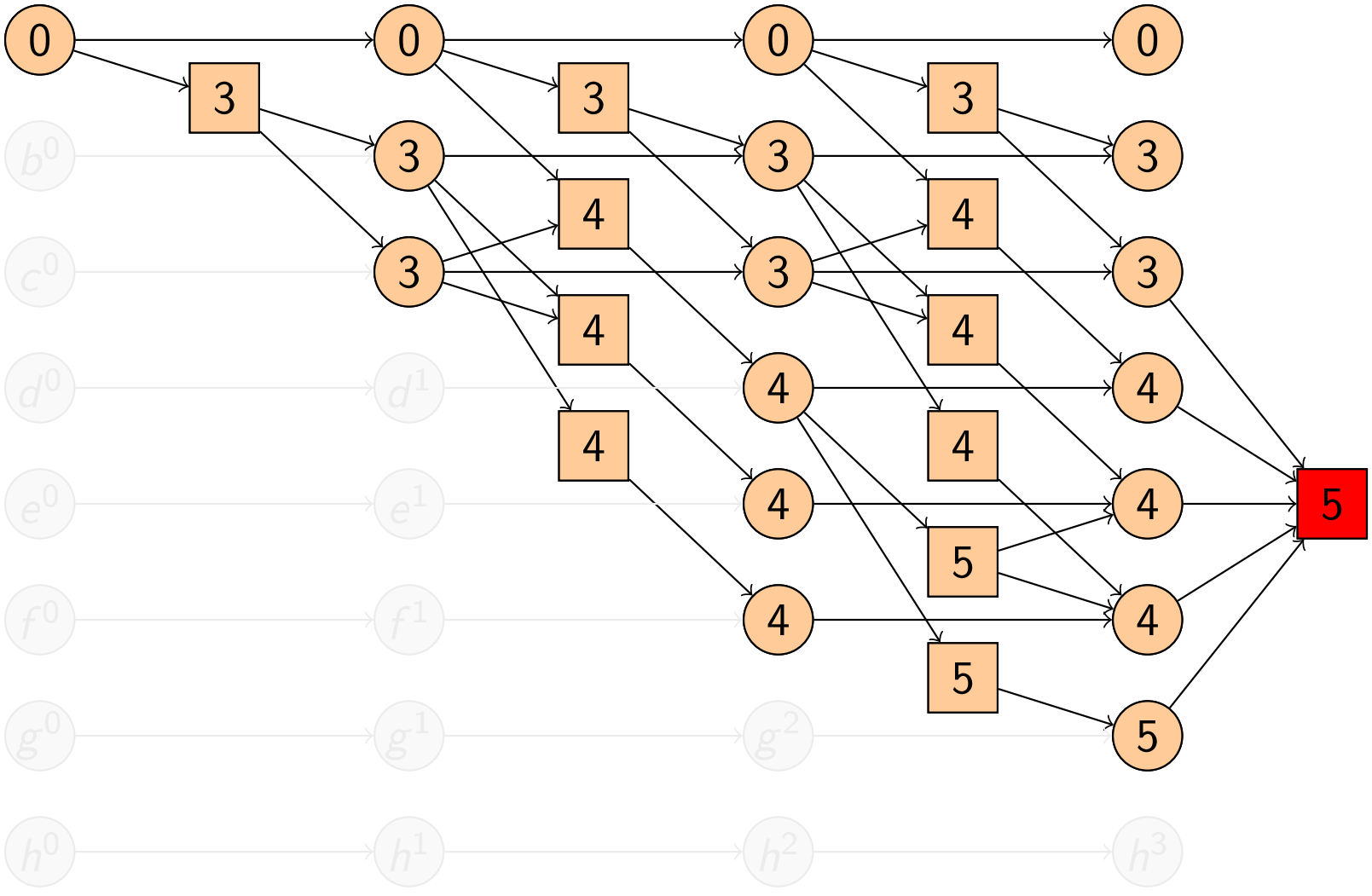
Illustrierendes Beispiel: $h^{\max}$



Illustrierendes Beispiel: $h^{\max}$

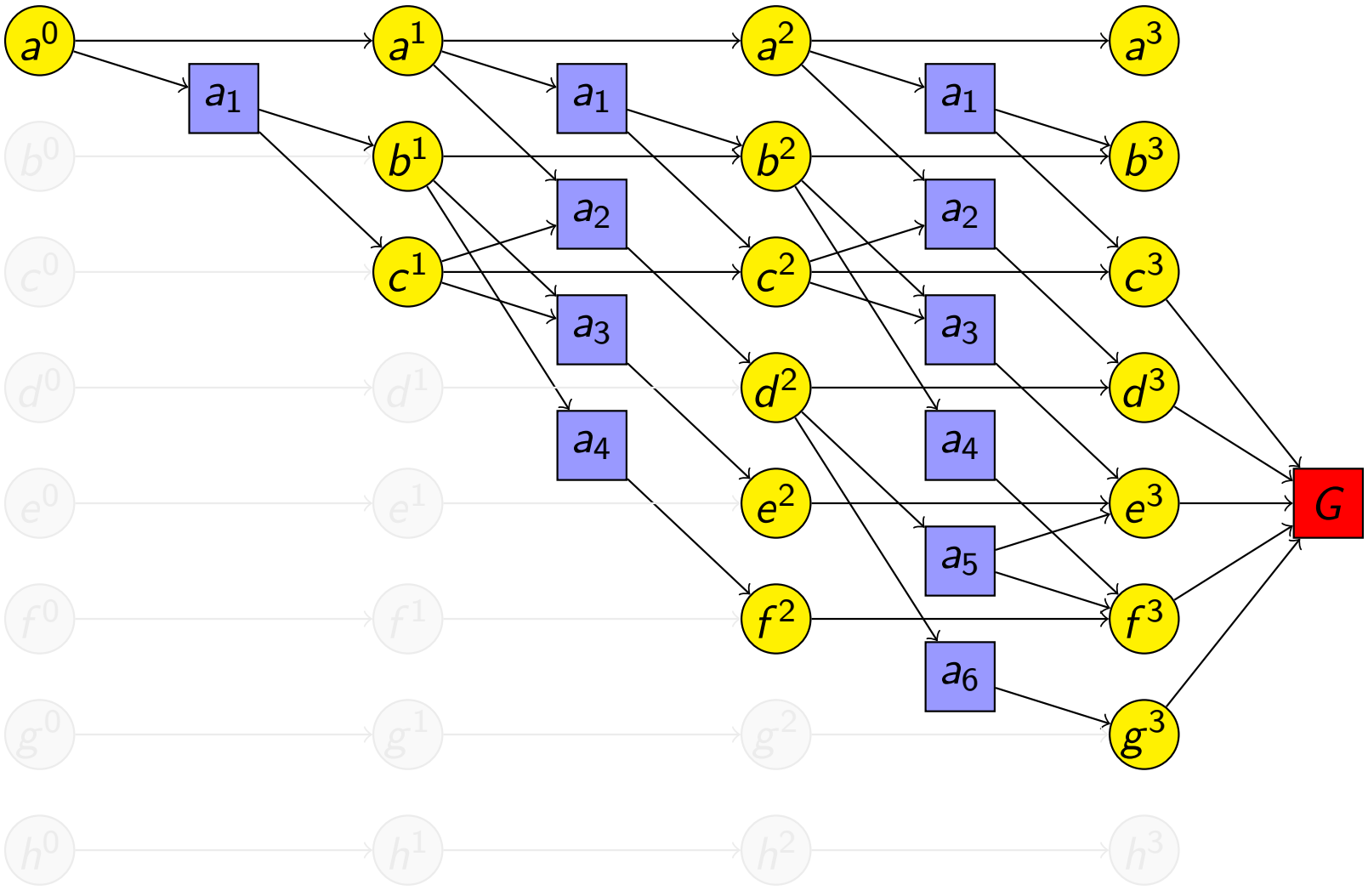


Illustrierendes Beispiel: $h^{\max}$

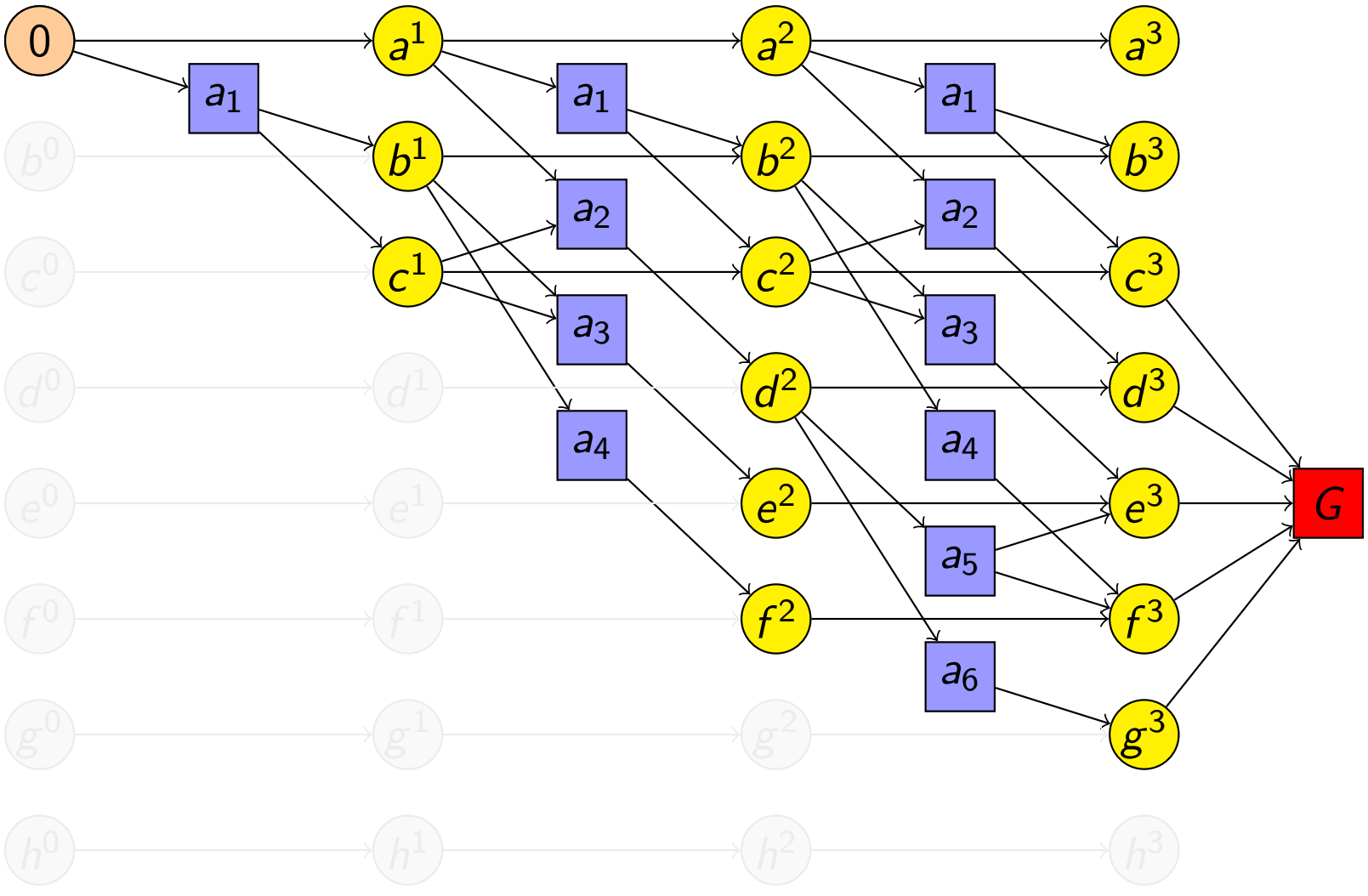


$$h^{\max}(\{a\}) = 5$$

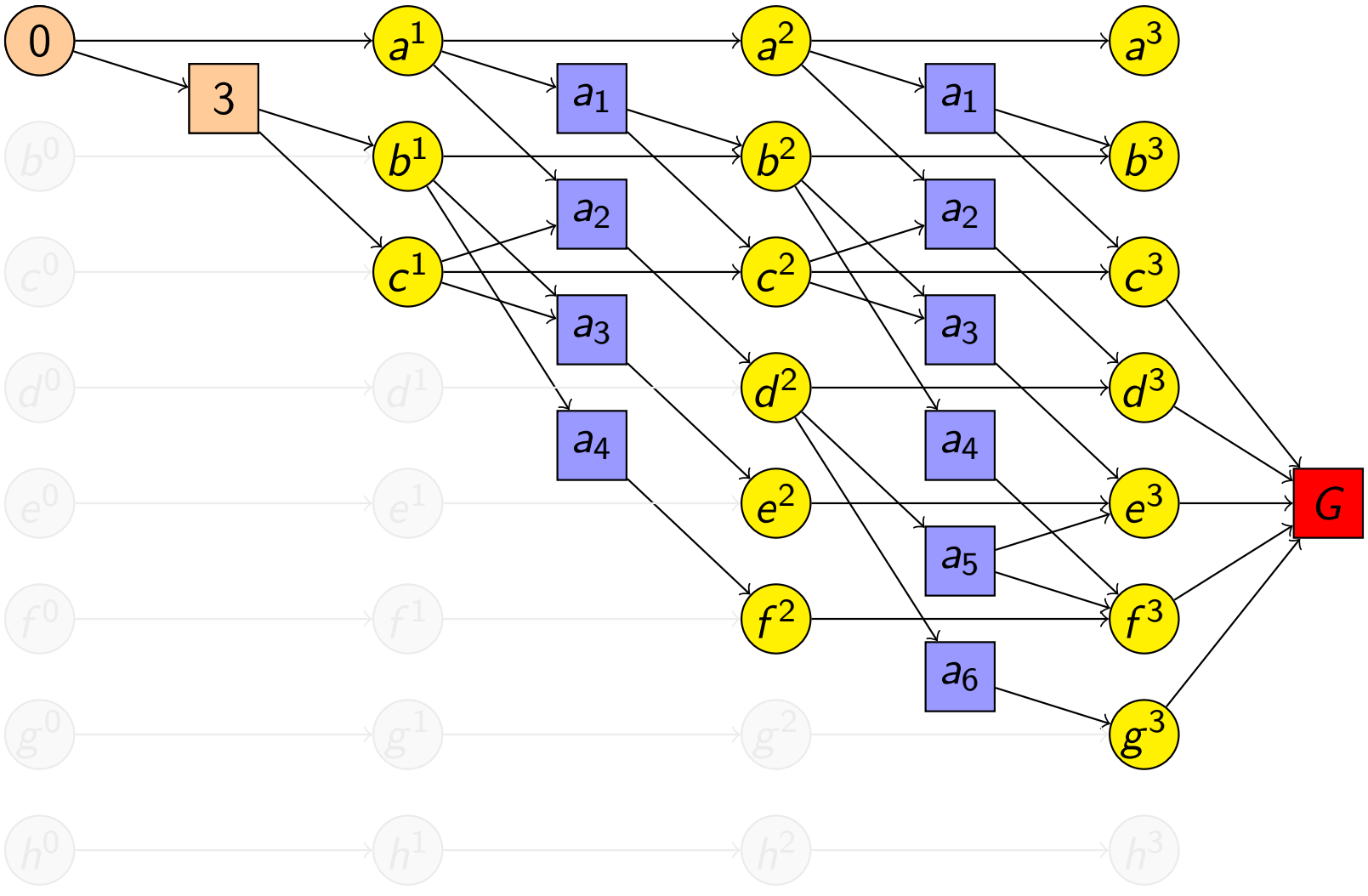
Illustrierendes Beispiel: h^{add}



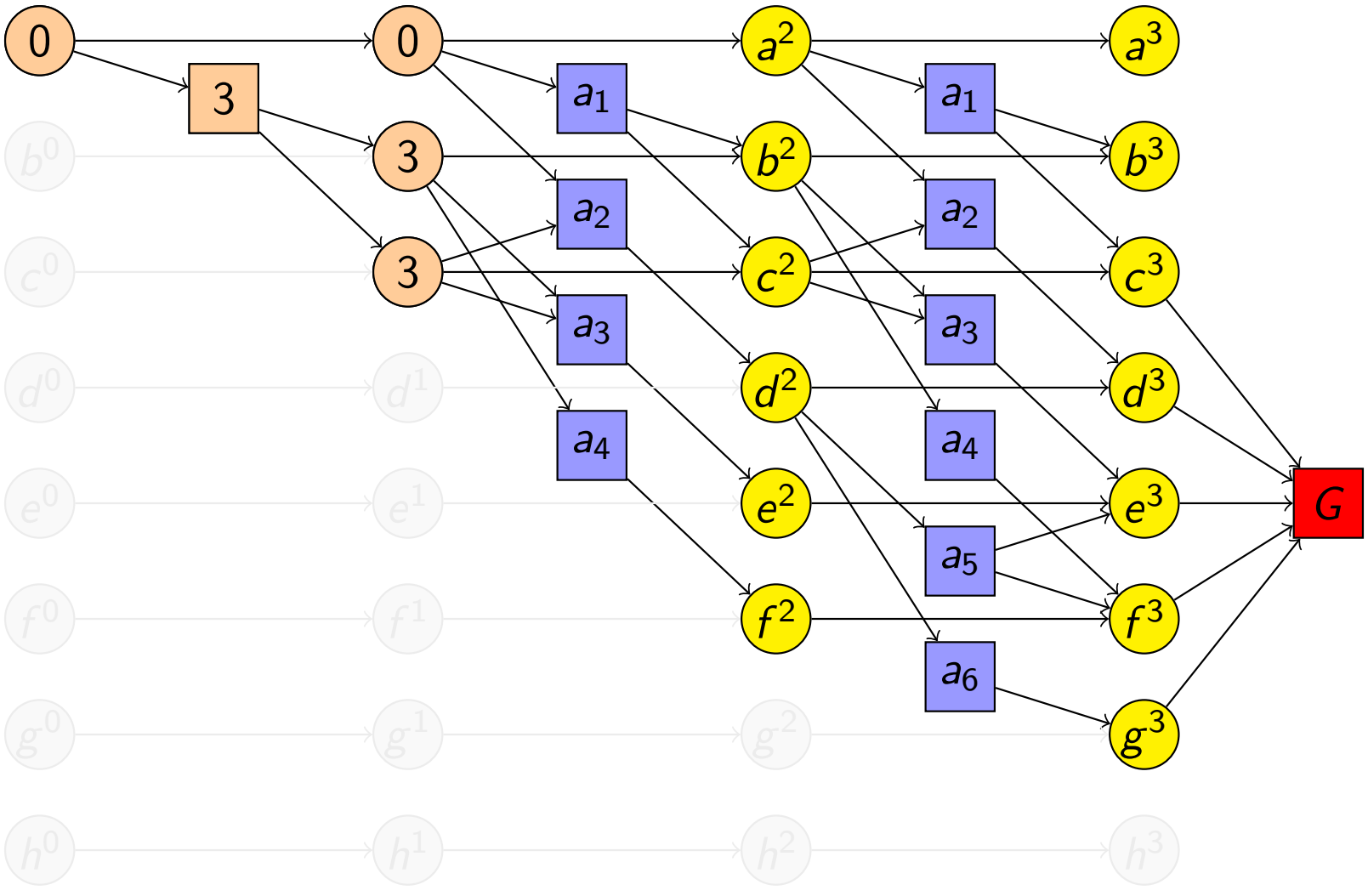
Illustrierendes Beispiel: h^{add}



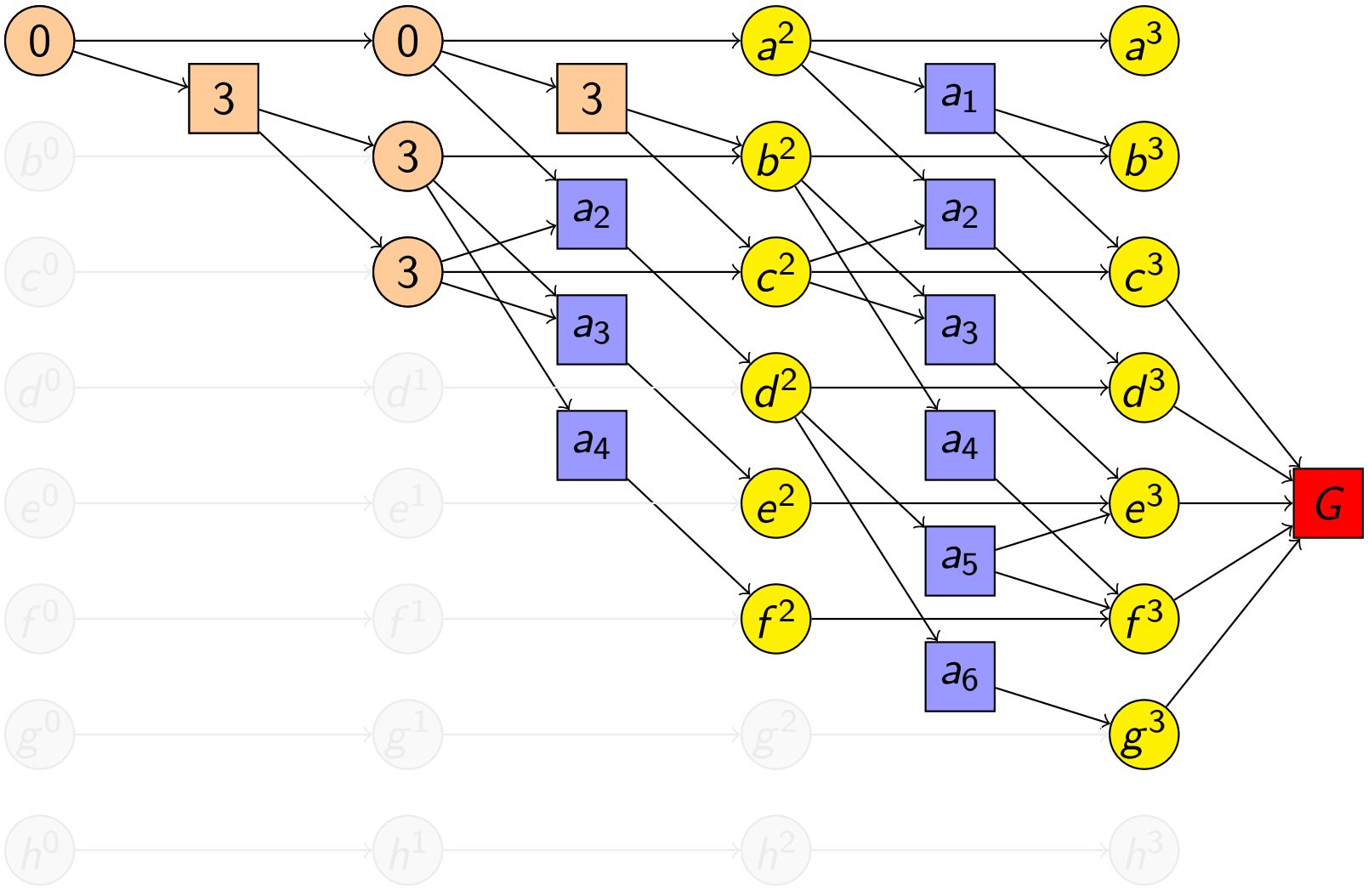
Illustrierendes Beispiel: h^{add}



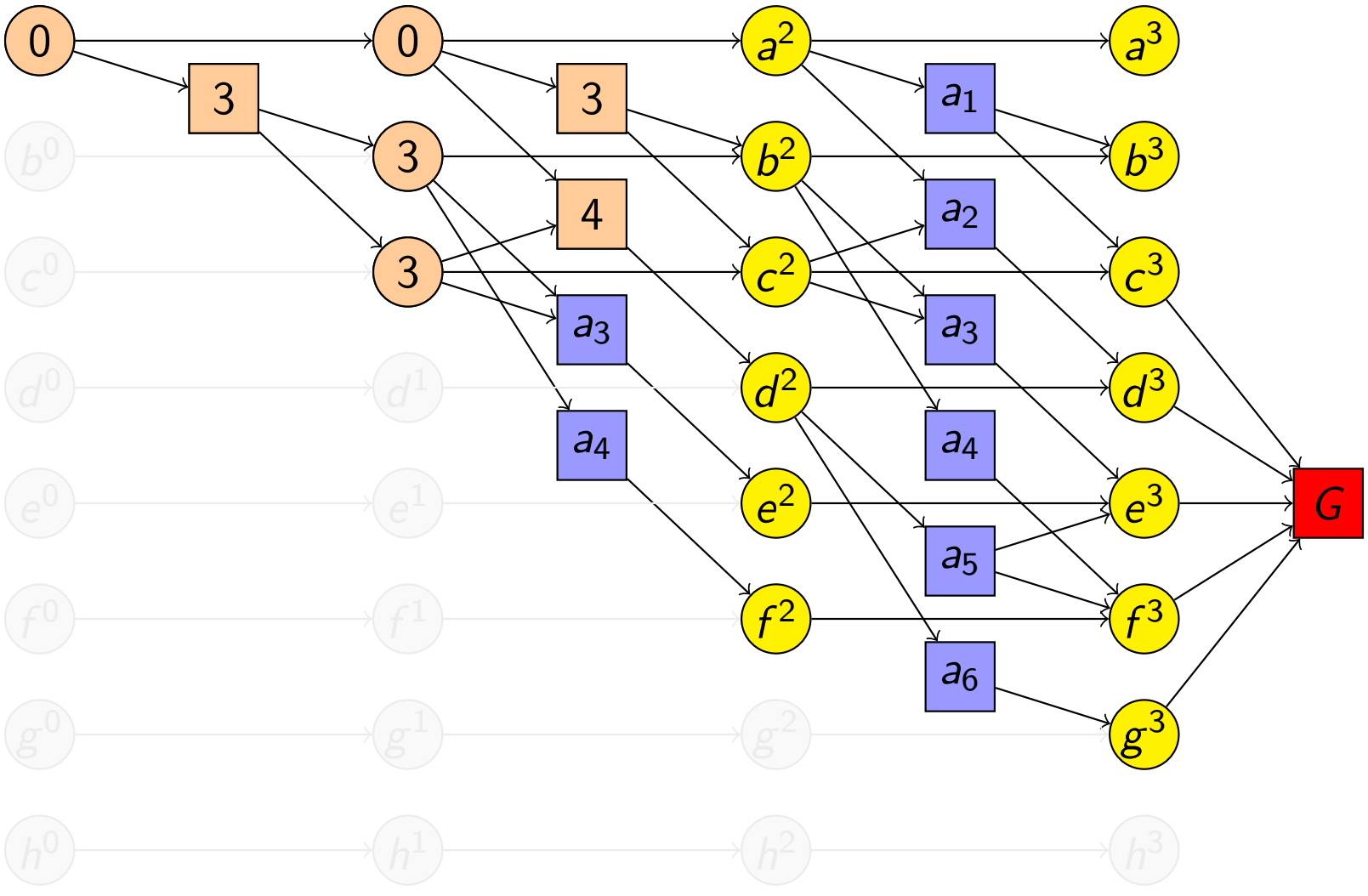
Illustrierendes Beispiel: h^{add}



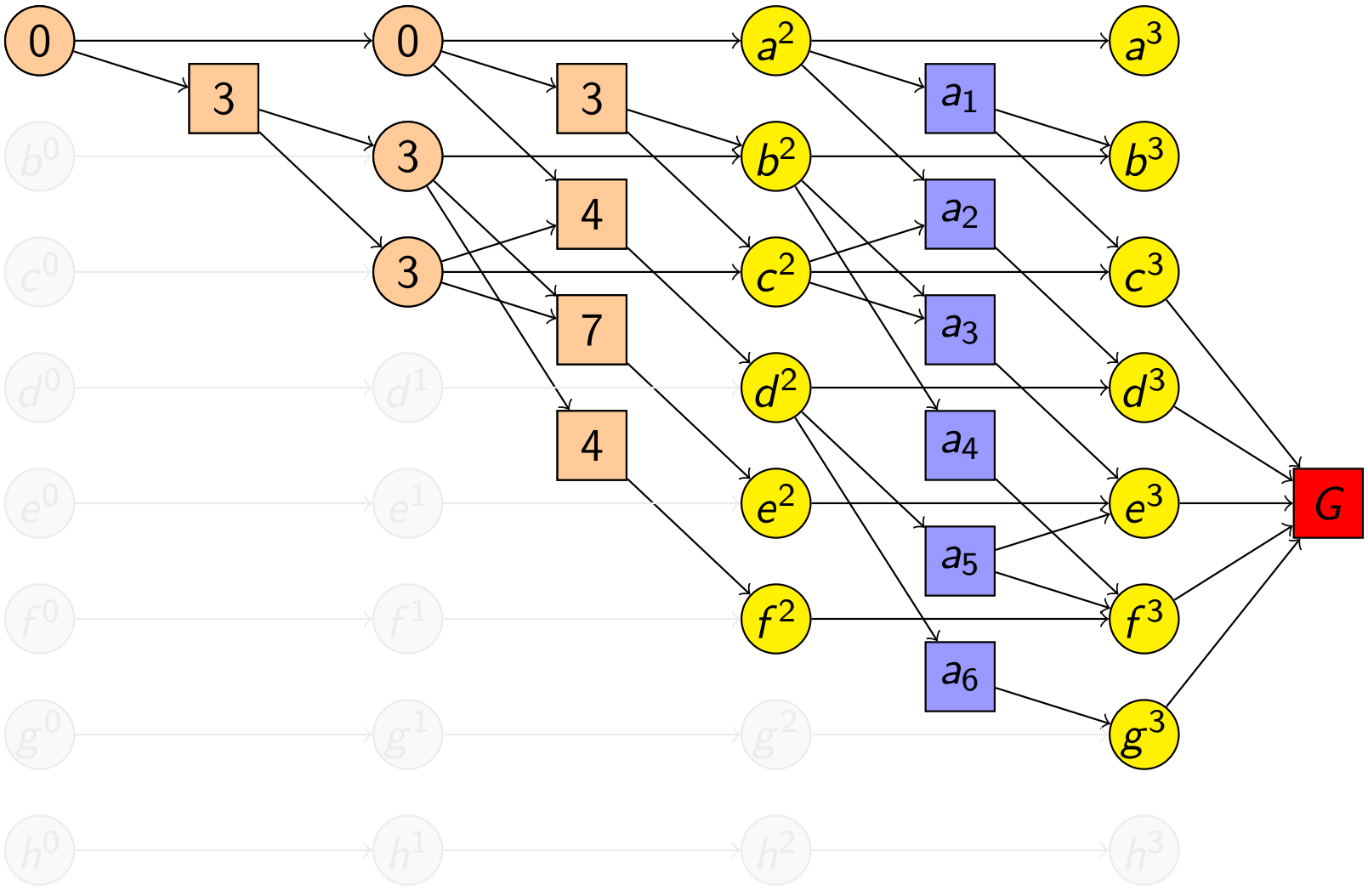
Illustrierendes Beispiel: h^{add}



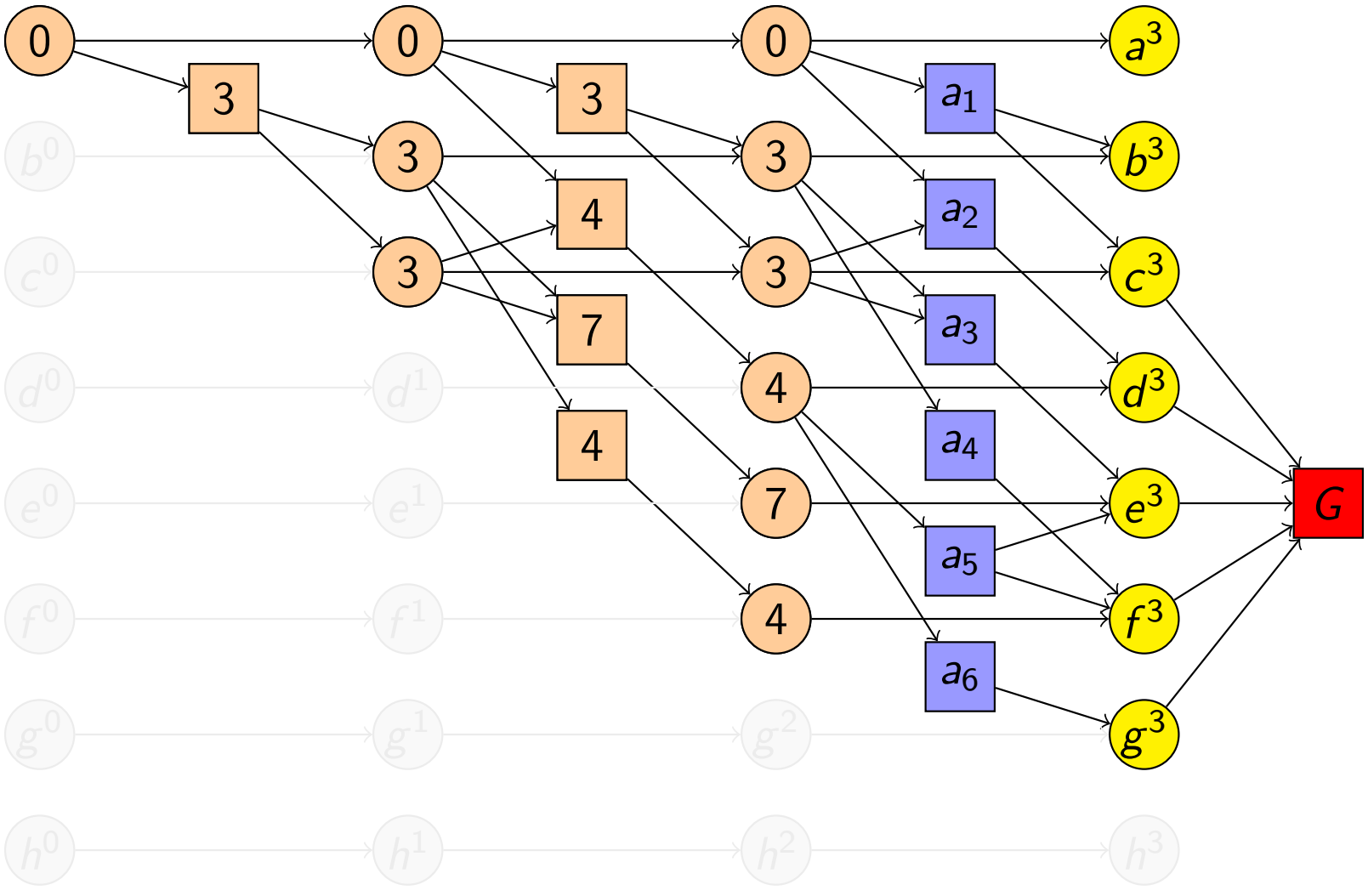
Illustrierendes Beispiel: h^{add}



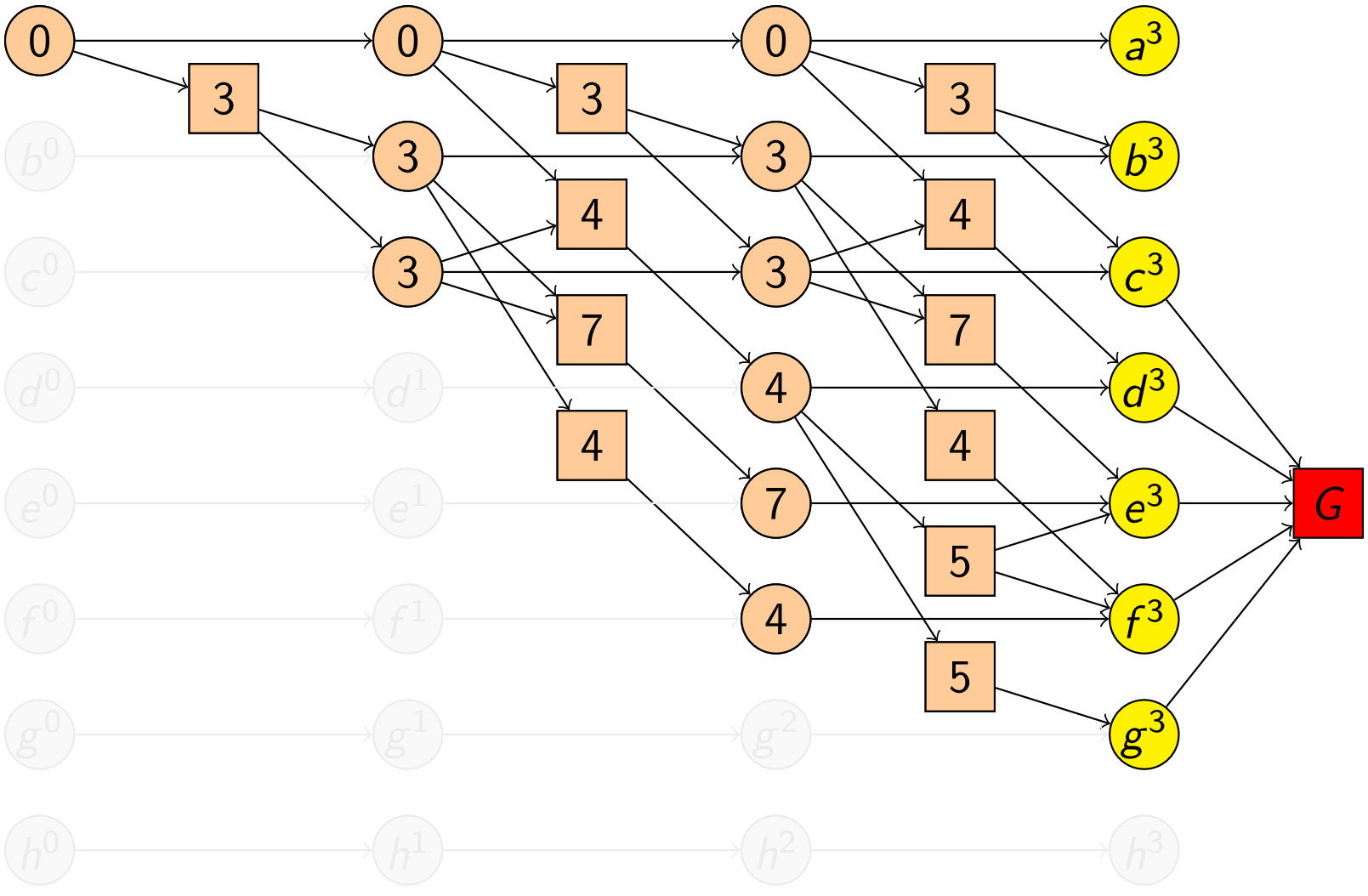
Illustrierendes Beispiel: h^{add}



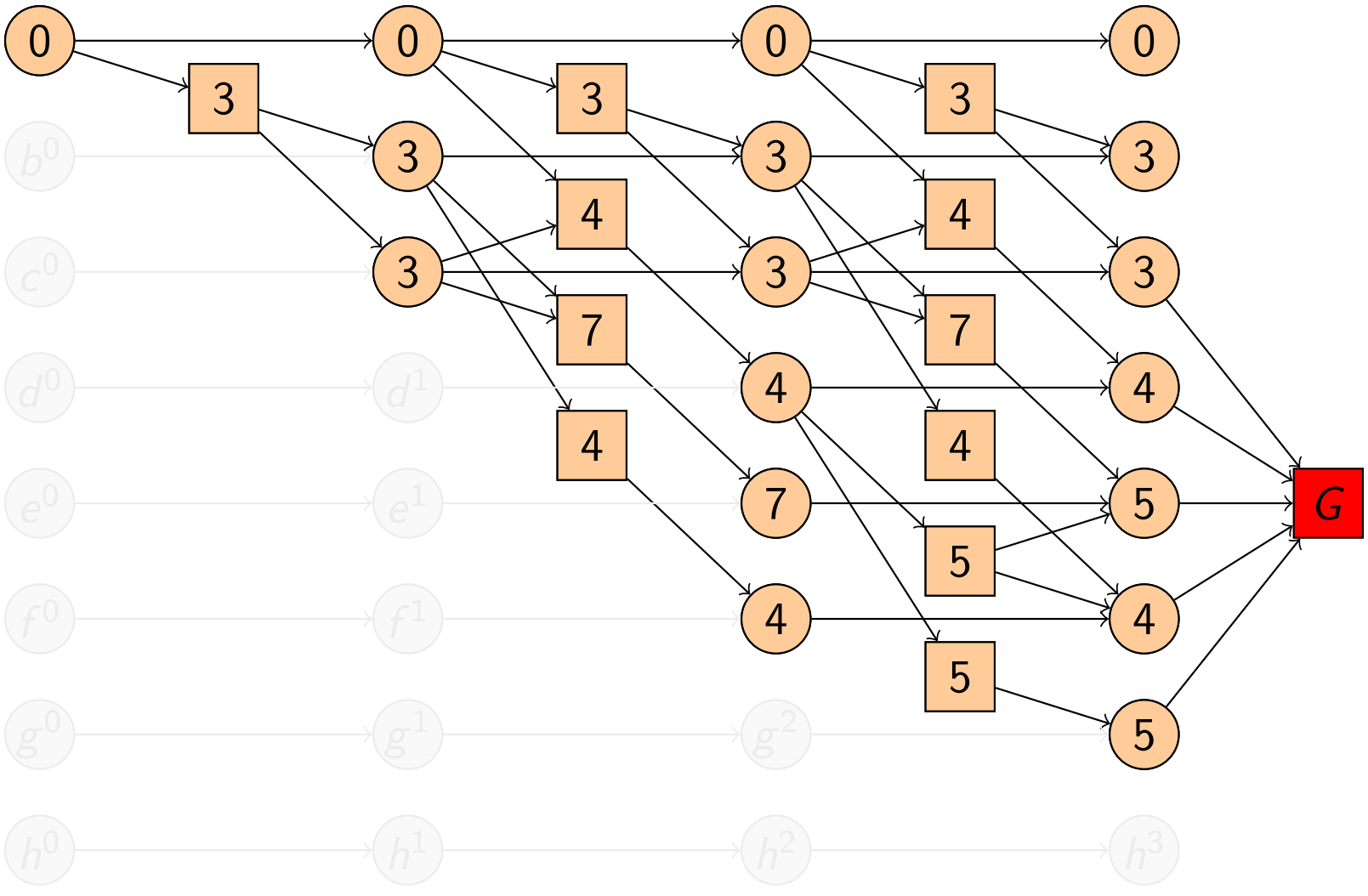
Illustrierendes Beispiel: h^{add}



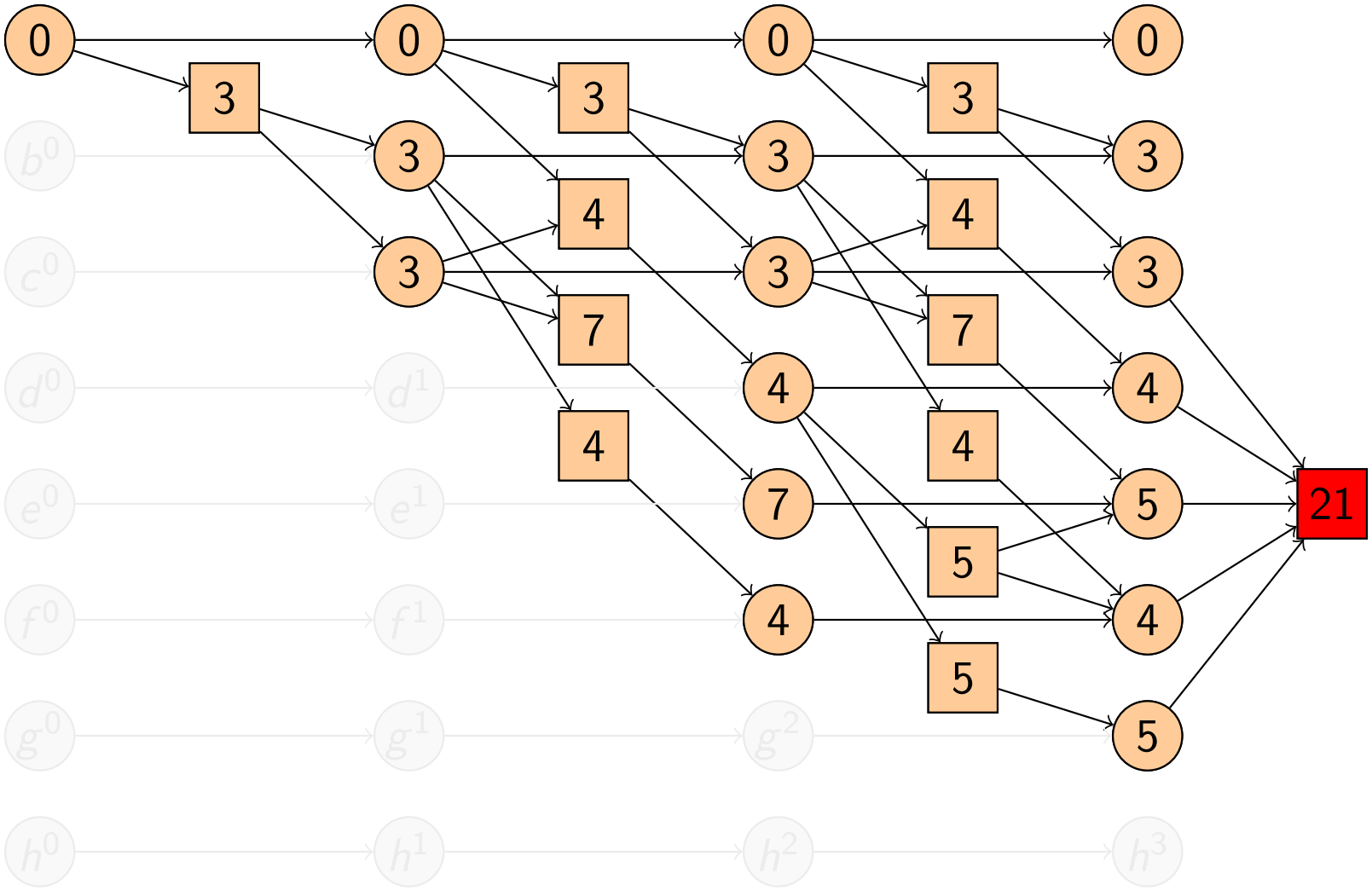
Illustrierendes Beispiel: h^{add}



Illustrierendes Beispiel: h^{add}



Illustrierendes Beispiel: h^{add}



$$h^{\text{add}}(\{a\}) = 21$$

Bemerkungen zu $h^{\max}$ und h^{add}

Vergleich von $h^{\max}$ und h^{add}:

- beide sind sicher und zielerkennend
- $h^{\max}$ ist zulässig und konsistent; h^{add} nicht

$\rightsquigarrow$ h^{add} nicht für optimale Handlungsplanung geeignet

Bemerkungen zu $h^{\max}$ und h^{add}

Vergleich von $h^{\max}$ und h^{add}:

- beide sind sicher und zielerkennend
- $h^{\max}$ ist zulässig und konsistent; h^{add} nicht

$\rightsquigarrow$ h^{add} nicht für optimale Handlungsplanung geeignet

- Dafür ist h^{add} meist sehr viel informativer als $h^{\max}$.
  Gierige Bestensuche mit h^{add} ist ein brauchbarer Algorithmus.

Bemerkungen zu $h^{\max}$ und h^{add}

Vergleich von $h^{\max}$ und h^{add}:

- beide sind sicher und zielerkennend
- $h^{\max}$ ist zulässig und konsistent; h^{add} nicht

$\rightsquigarrow$ h^{add} nicht für optimale Handlungsplanung geeignet

- Dafür ist h^{add} meist sehr viel informativer als $h^{\max}$.
  Gierige Bestensuche mit h^{add} ist ein brauchbarer Algorithmus.
- Neben der mangelnden Zulässigkeit ist ein Problem von h^{add},
  dass es oft sehr stark die Kosten überschätzt,
  weil es positive Synergien zwischen Teilzielen nicht erkennt.

Bemerkungen zu $h^{\max}$ und h^{add}

Vergleich von $h^{\max}$ und h^{add}:

- beide sind sicher und zielerkennend
- $h^{\max}$ ist zulässig und konsistent; h^{add} nicht

$\rightsquigarrow$ h^{add} nicht für optimale Handlungsplanung geeignet

- Dafür ist h^{add} meist sehr viel informativer als $h^{\max}$.
  Gierige Bestensuche mit h^{add} ist ein brauchbarer Algorithmus.
- Neben der mangelnden Zulässigkeit ist ein Problem von h^{add},
  dass es oft sehr stark die Kosten überschätzt,
  weil es positive Synergien zwischen Teilzielen nicht erkennt.

$\rightsquigarrow$ FF-Heuristik

FF-Heuristik

Die FF-Heuristik

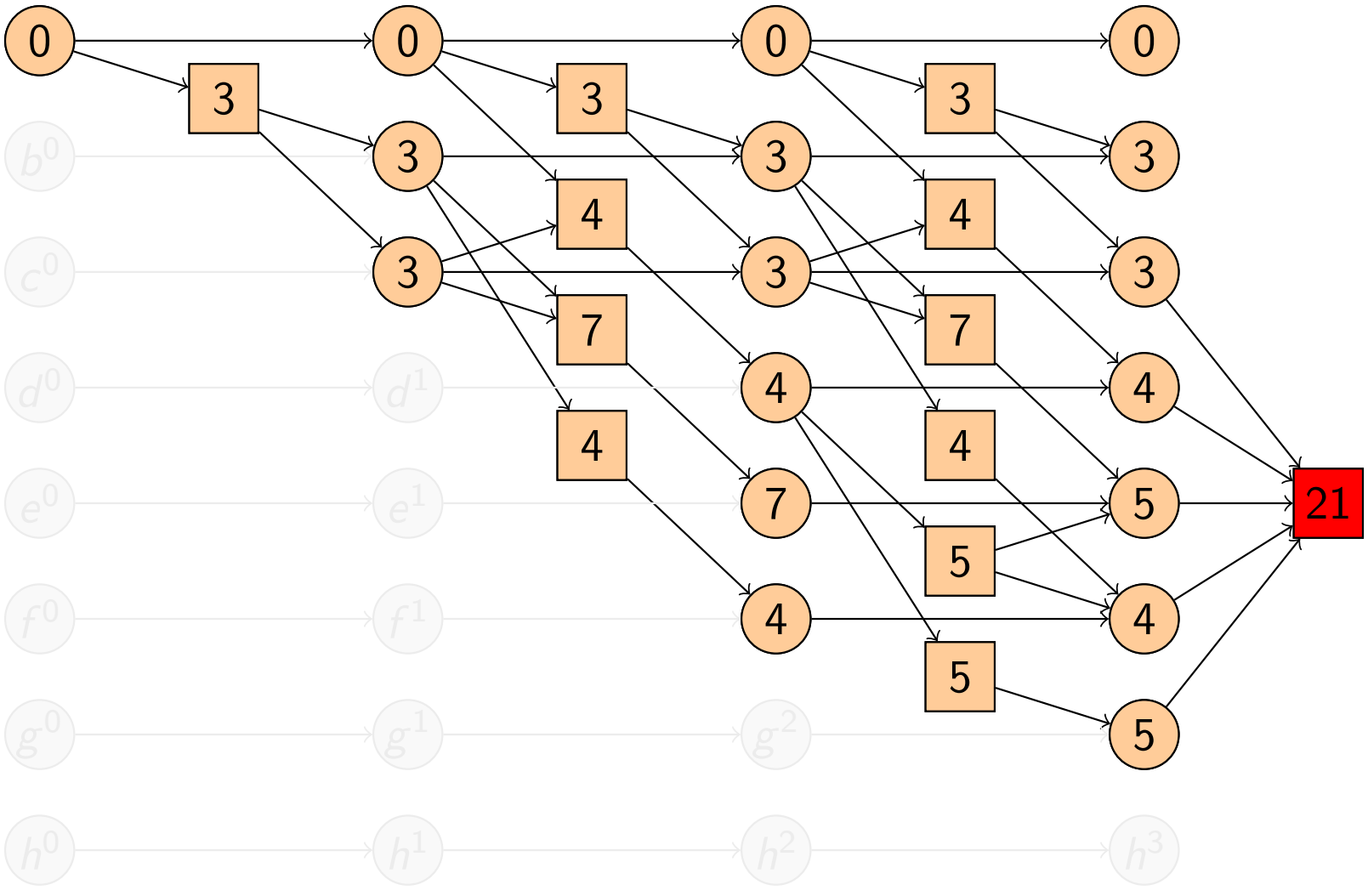
identisch zu h^{add}, aber zusätzliche Schritte am Ende:

- Markiere Zielknoten der letzten Graphebene.
- Wende folgende Markierungsregeln an, bis nichts mehr zu tun:
  - Aktions- oder Zielknoten markiert?
    $\rightsquigarrow$ markiere alle Vorgänger
  - Variablenknoten v^i in Ebene $i \geq 1$ markiert?
    $\rightsquigarrow$ markiere einen Vorgänger mit minimalem h^{add}-Wert
    (Tie-Breaking: bevorzuge Variablenknoten; sonst beliebig)

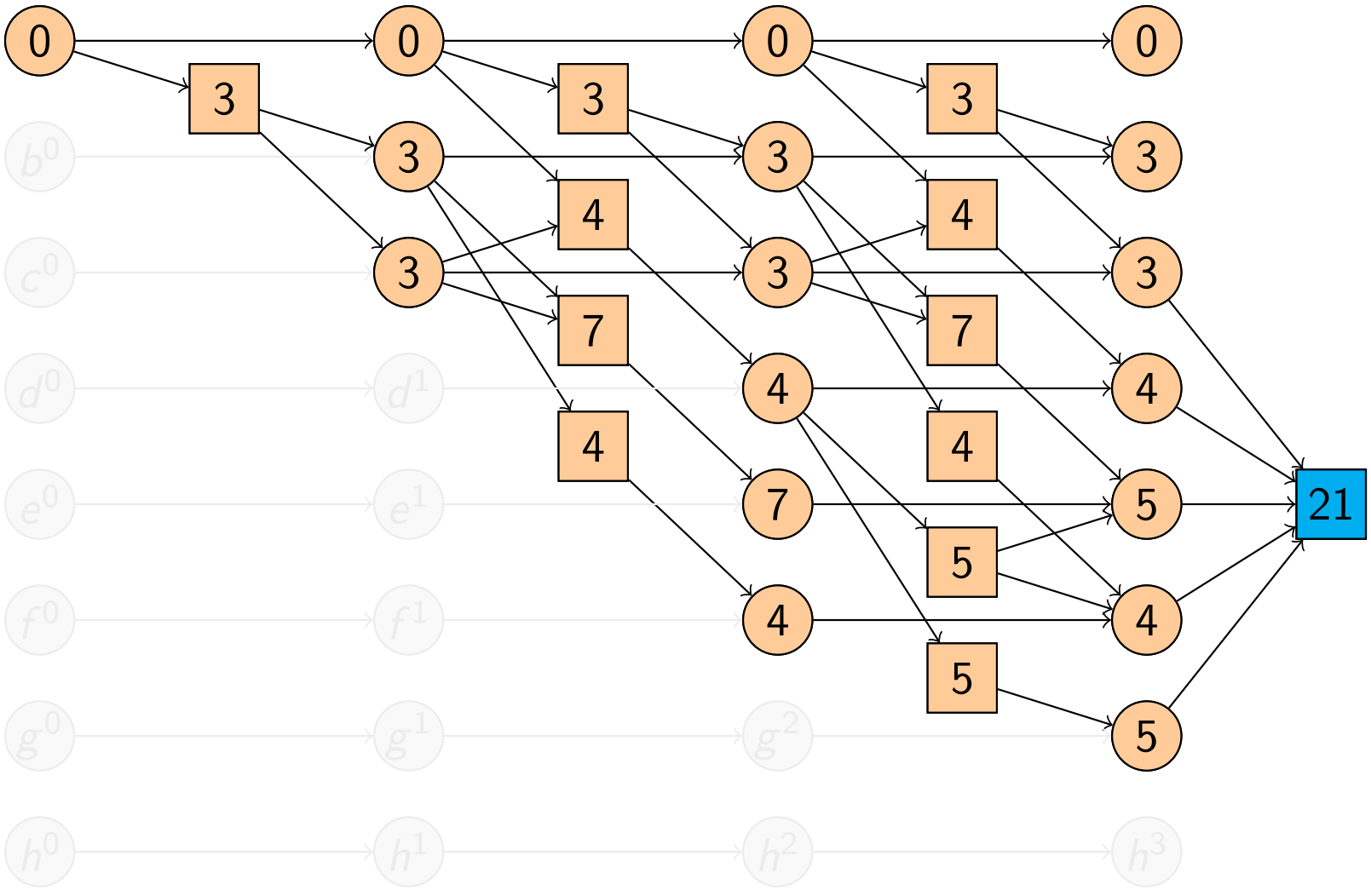
Heuristikwert:

- Die zu den markierten Aktionsknoten gehörigen Aktionen bilden einen relaxierten Plan.
- Die Kosten dieses Plans bilden den Heuristikwert.

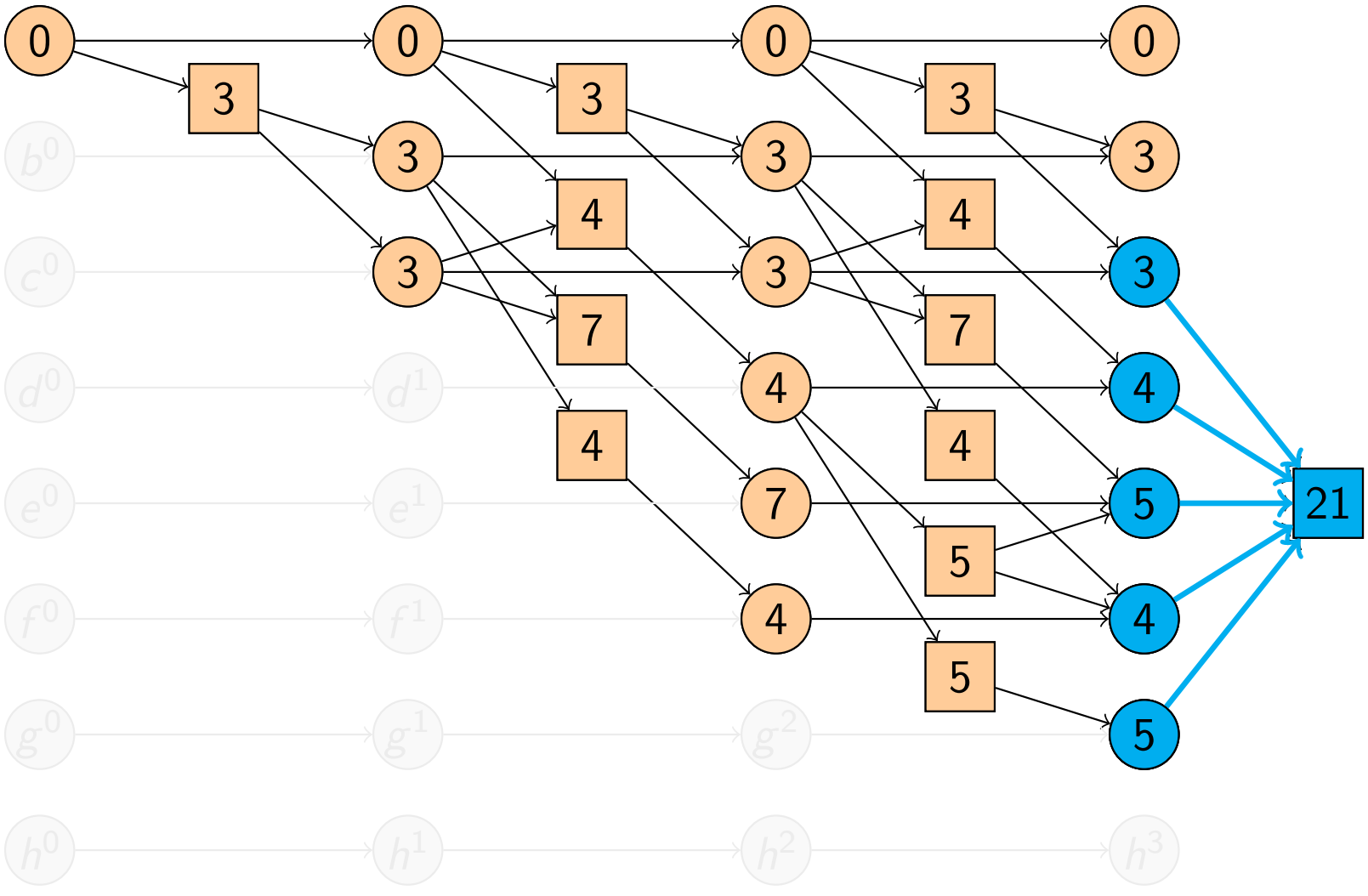
Illustrierendes Beispiel: h^{FF}



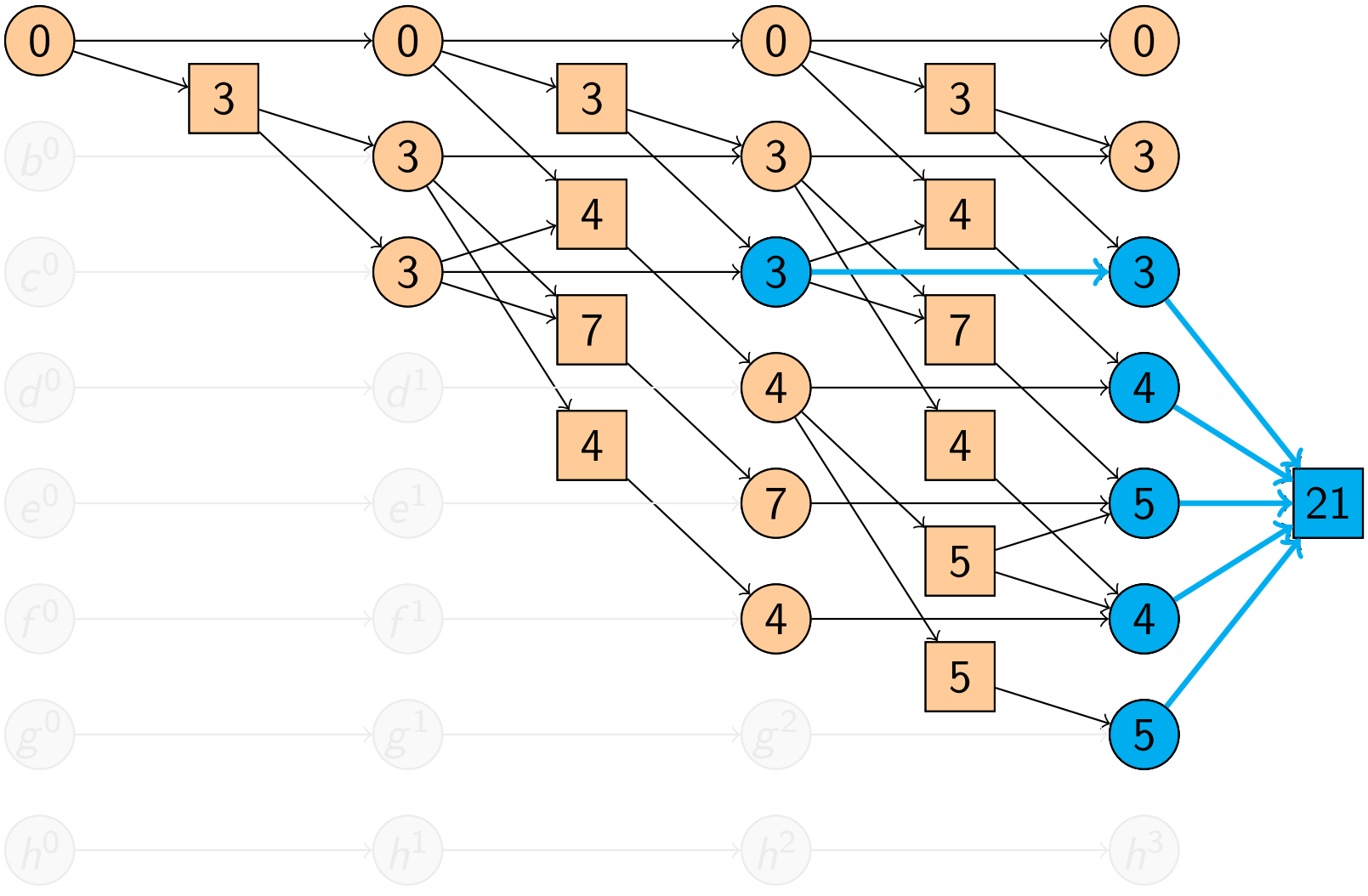
Illustrierendes Beispiel: h^{FF}



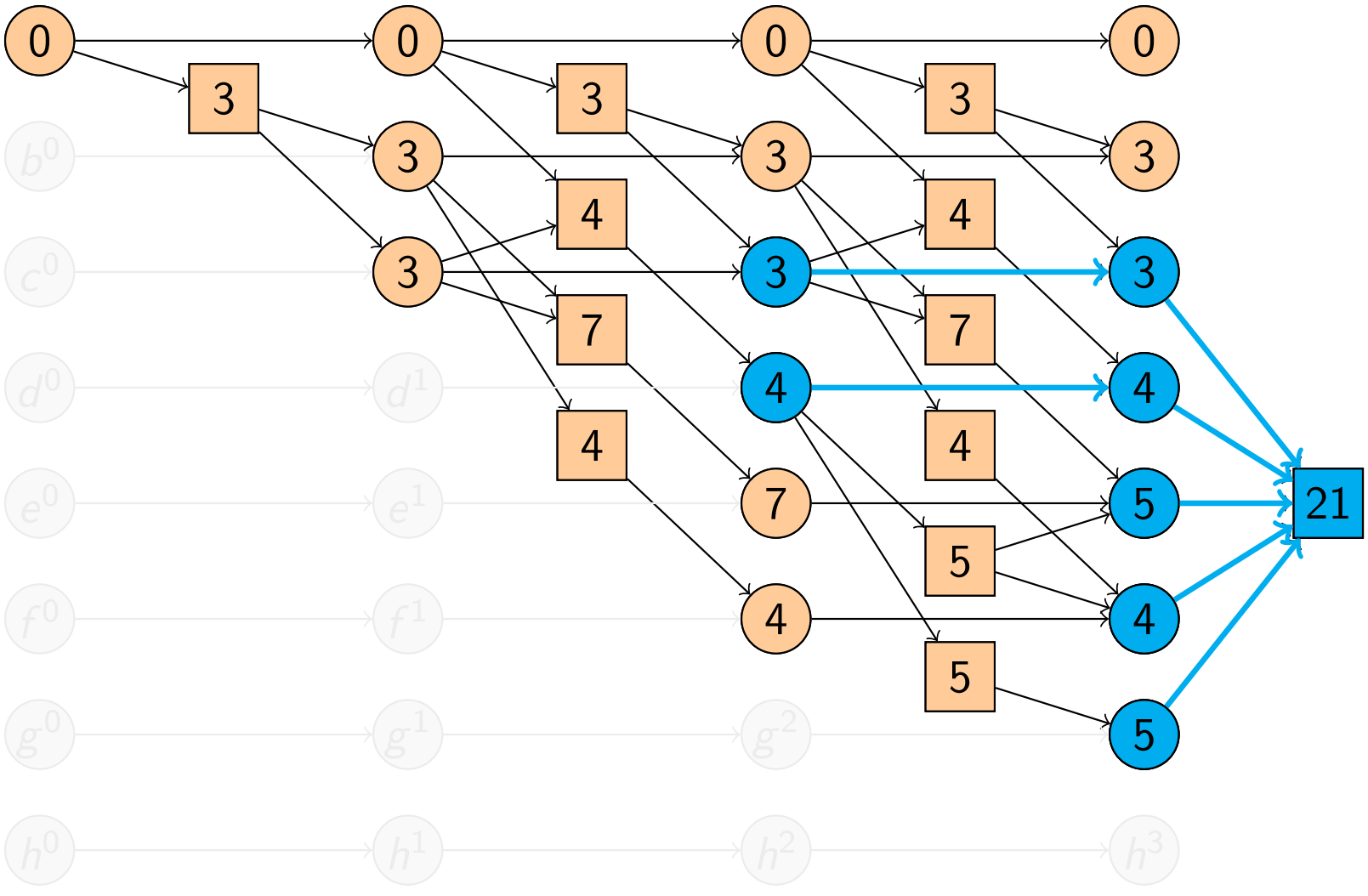
Illustrierendes Beispiel: h^{FF}



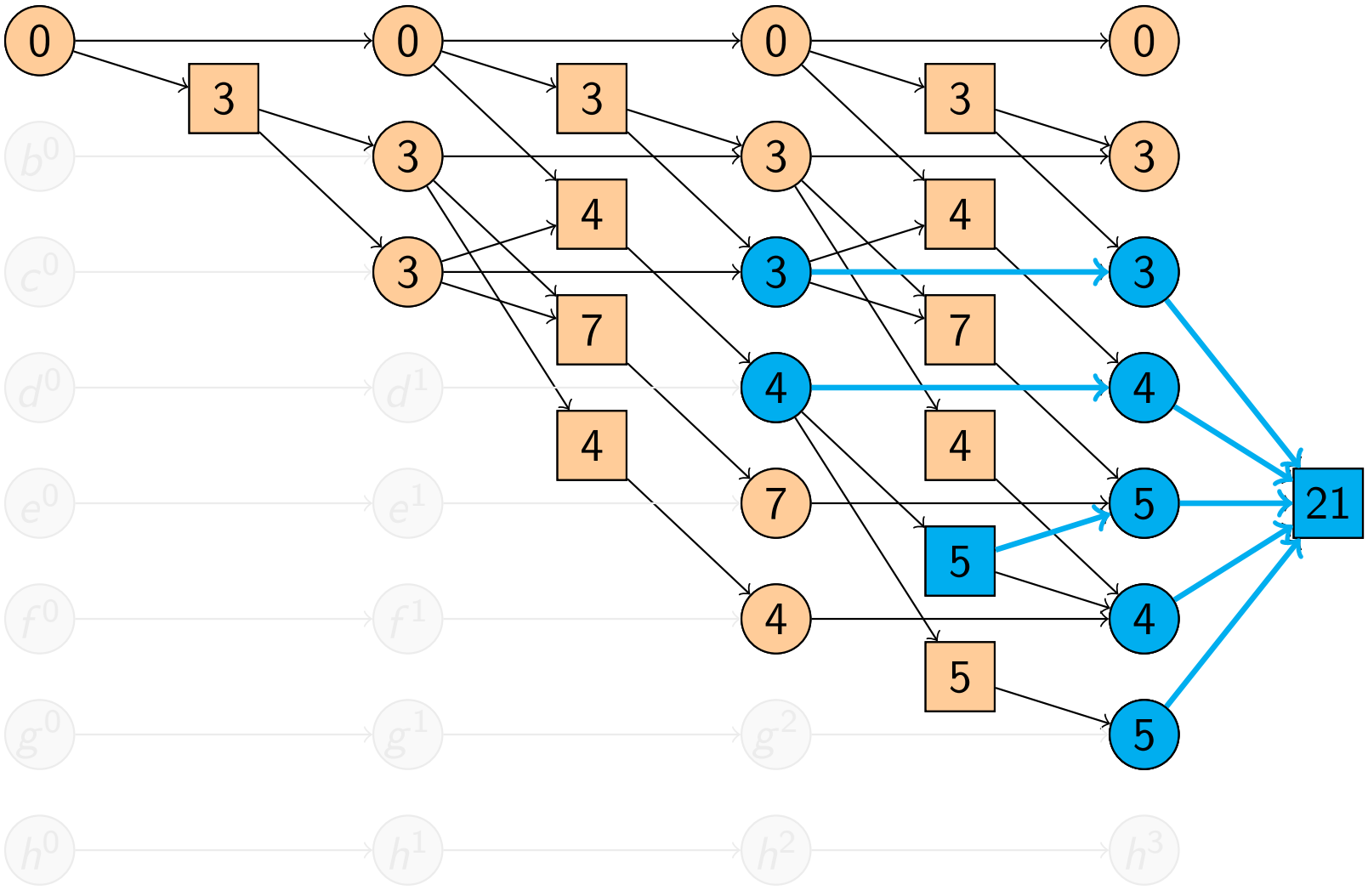
Illustrierendes Beispiel: h^{FF}



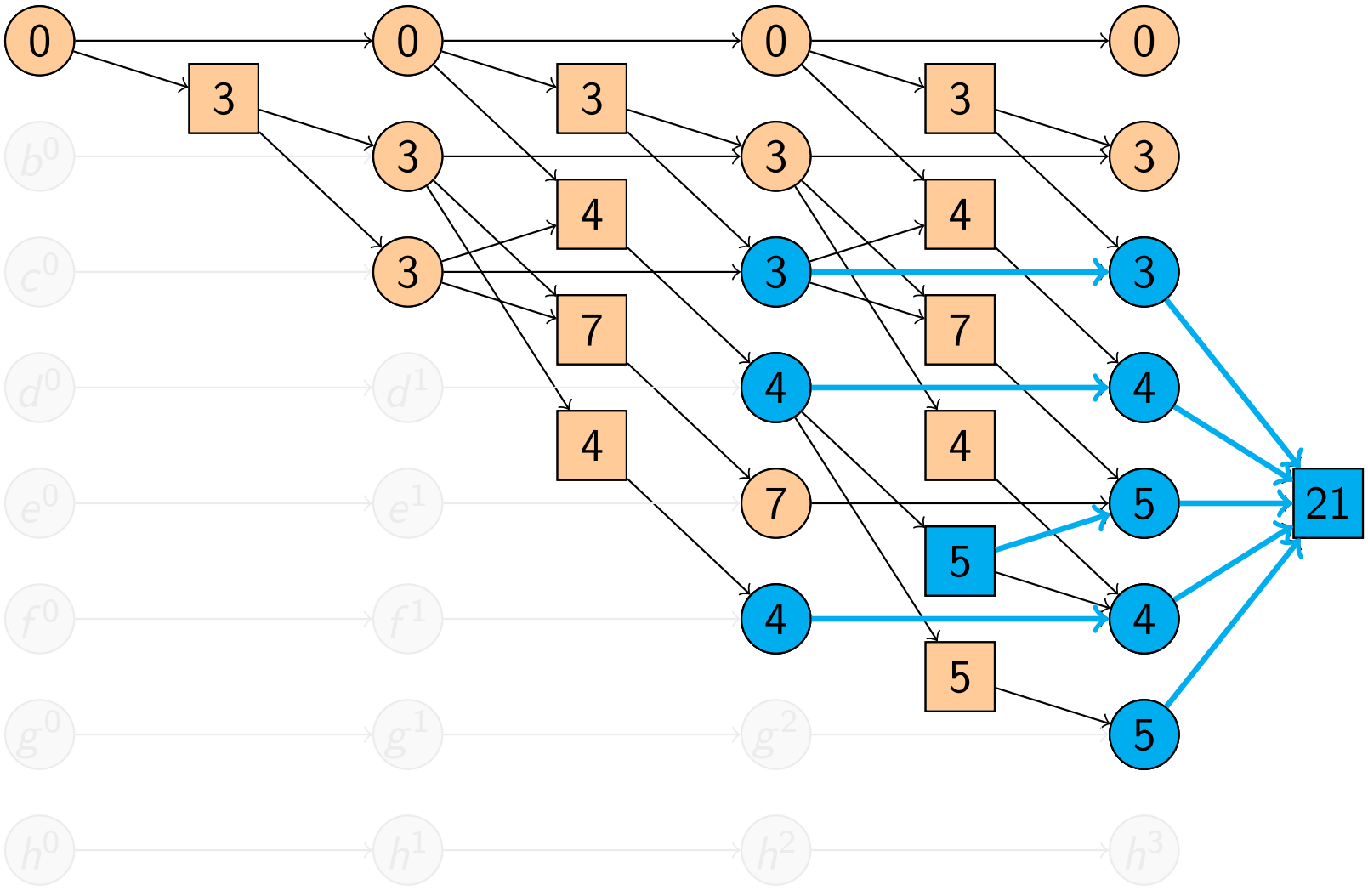
Illustrierendes Beispiel: h^{FF}



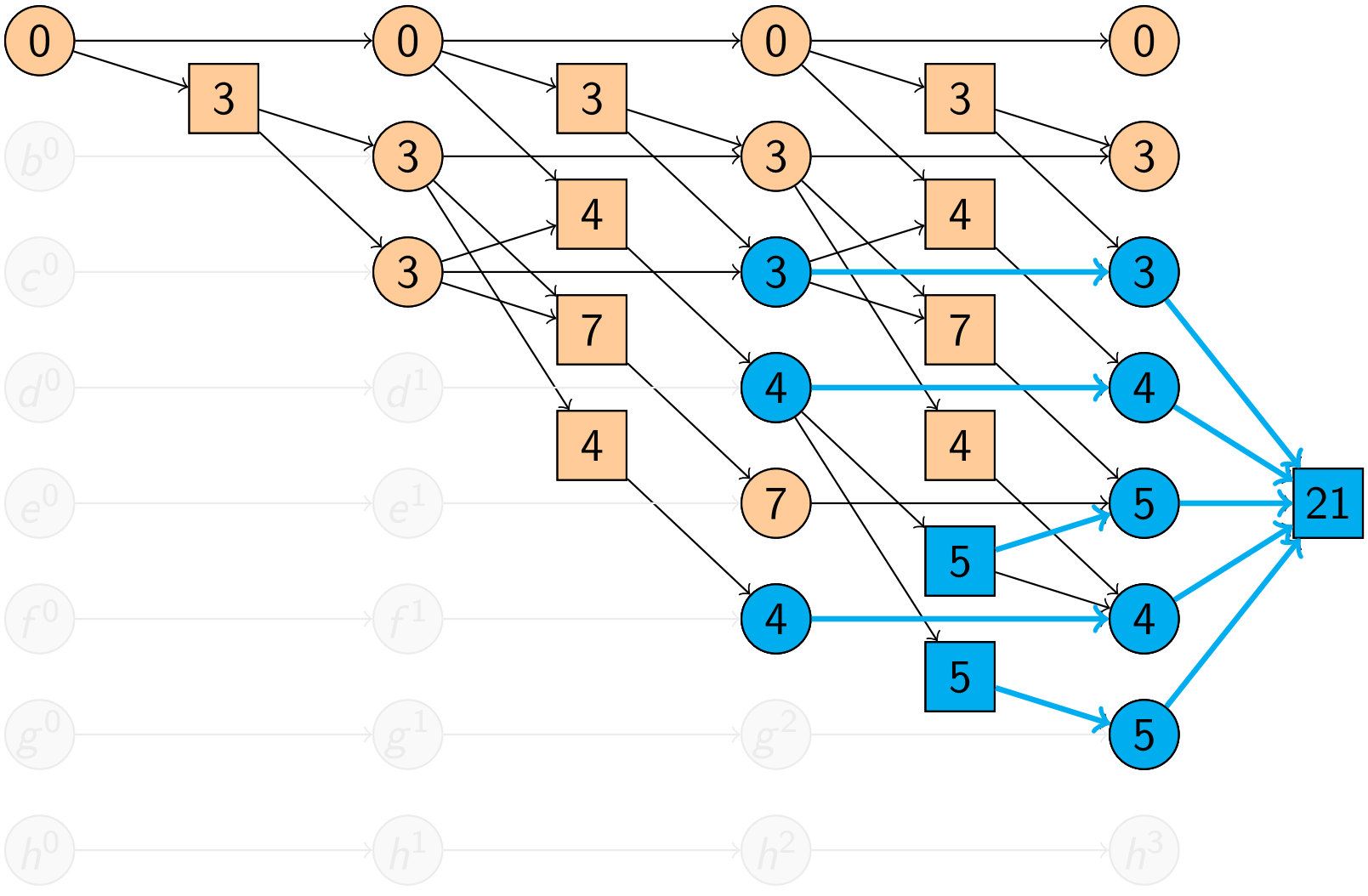
Illustrierendes Beispiel: h^{FF}



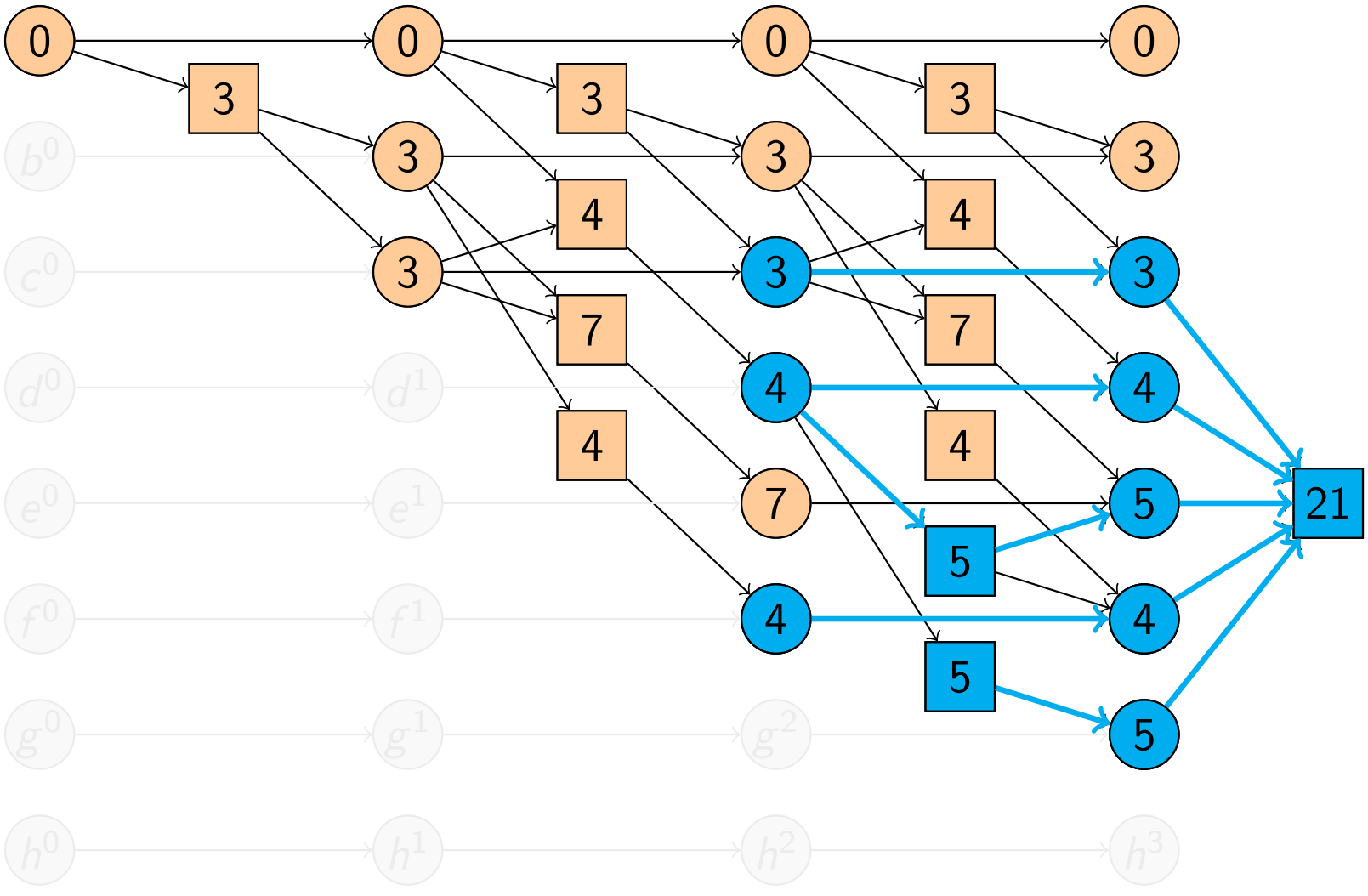
Illustrierendes Beispiel: h^{FF}



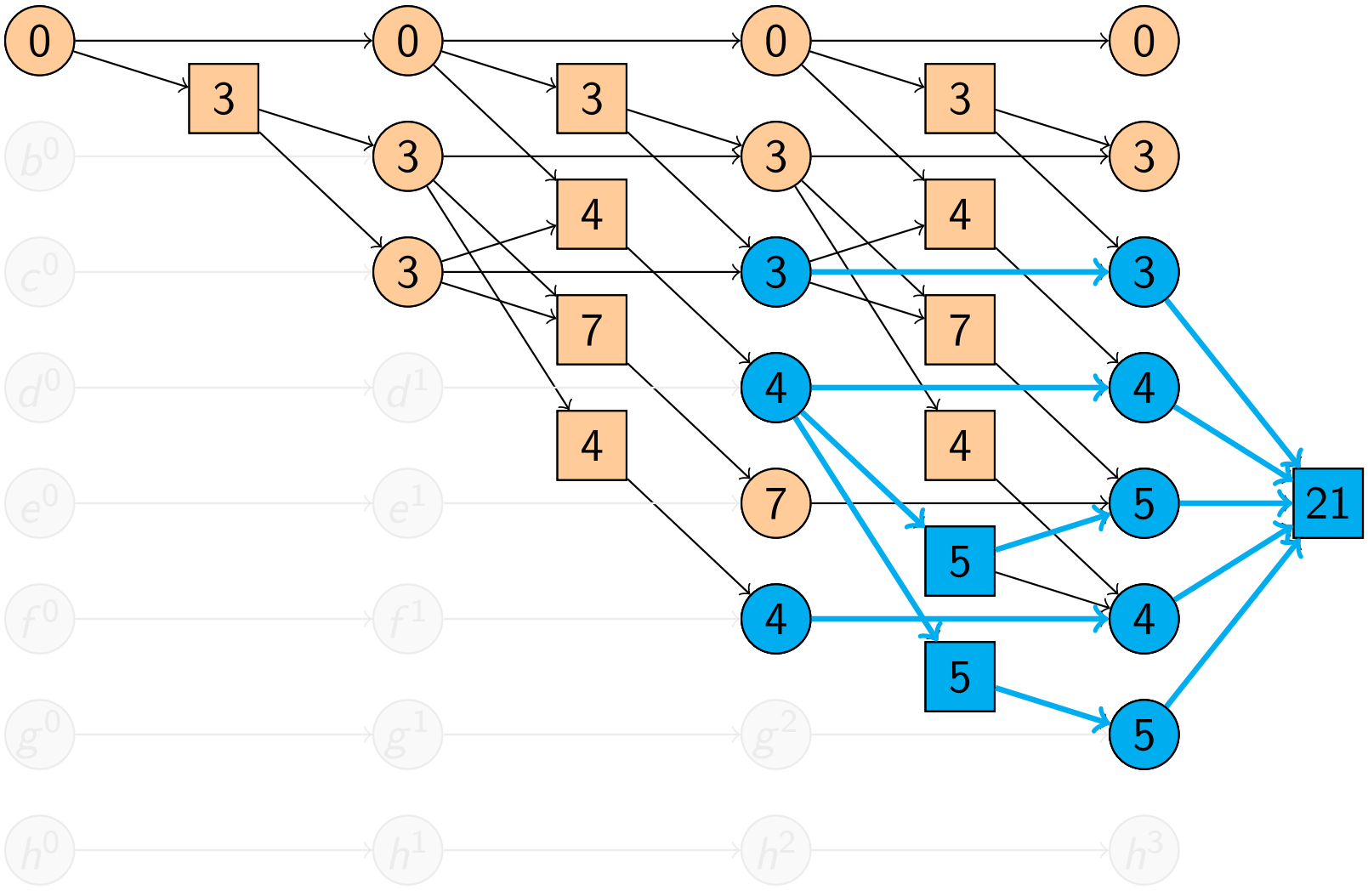
Illustrierendes Beispiel: h^{FF}



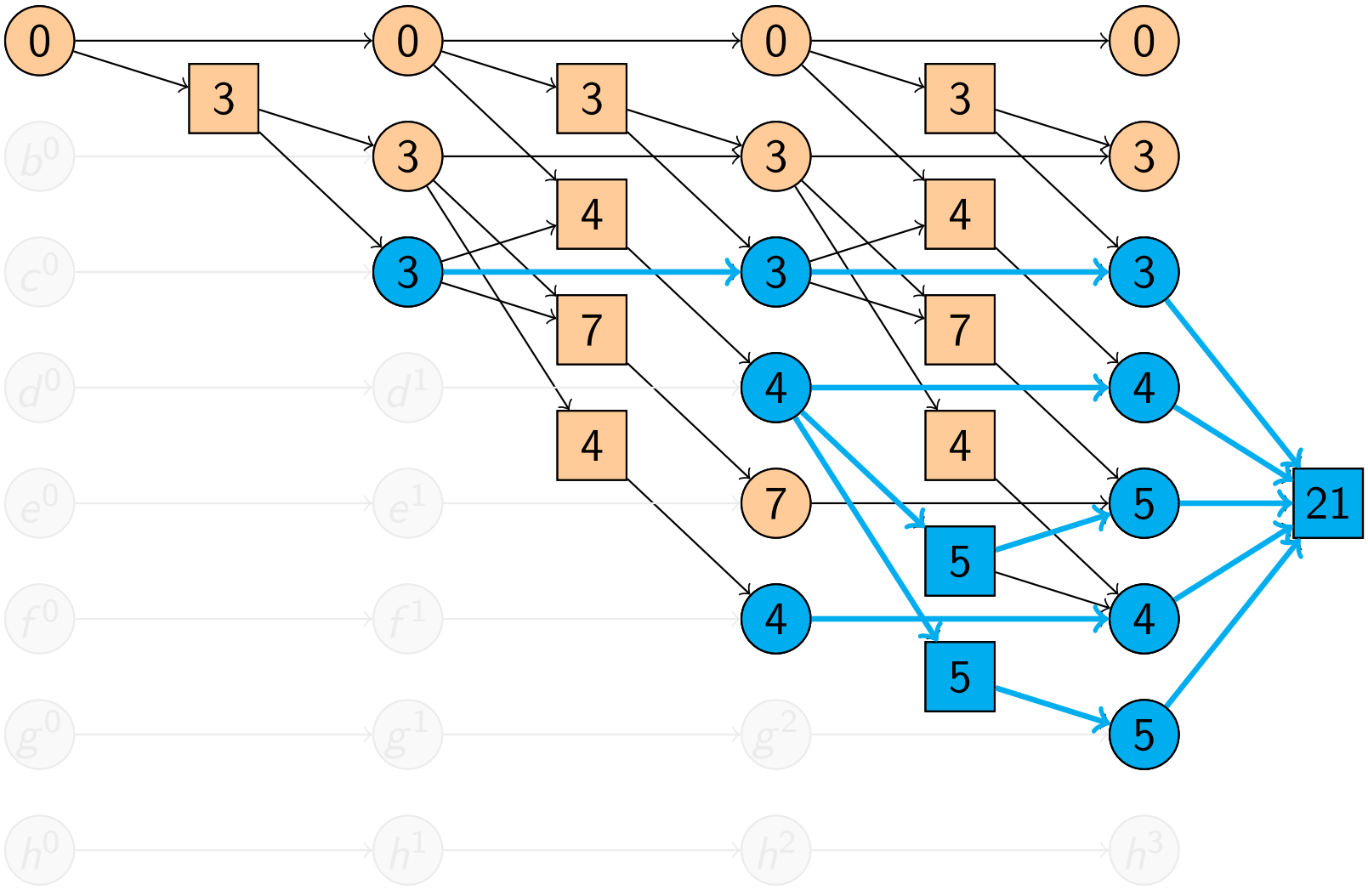
Illustrierendes Beispiel: h^{FF}



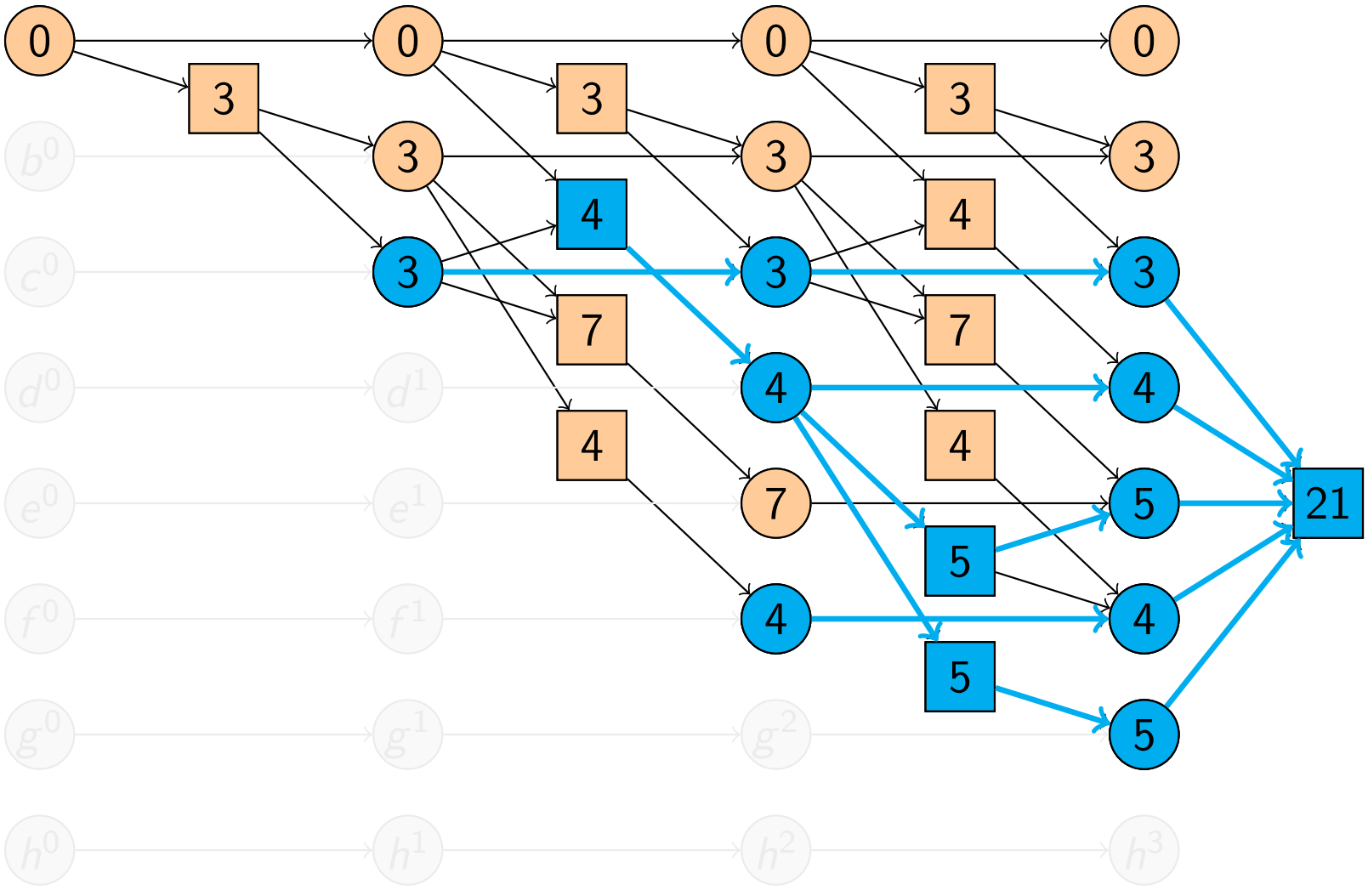
Illustrierendes Beispiel: h^{FF}



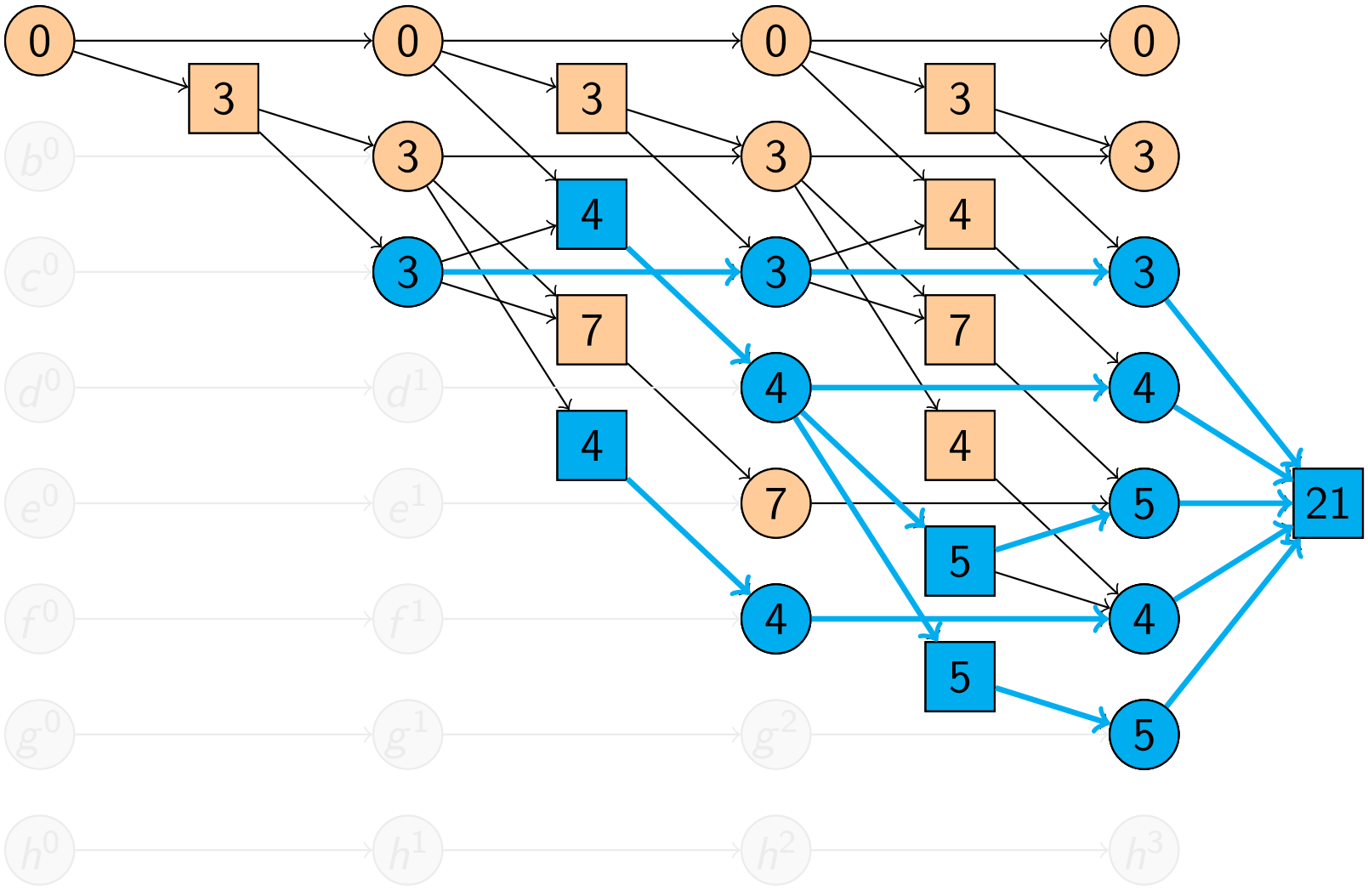
Illustrierendes Beispiel: h^{FF}



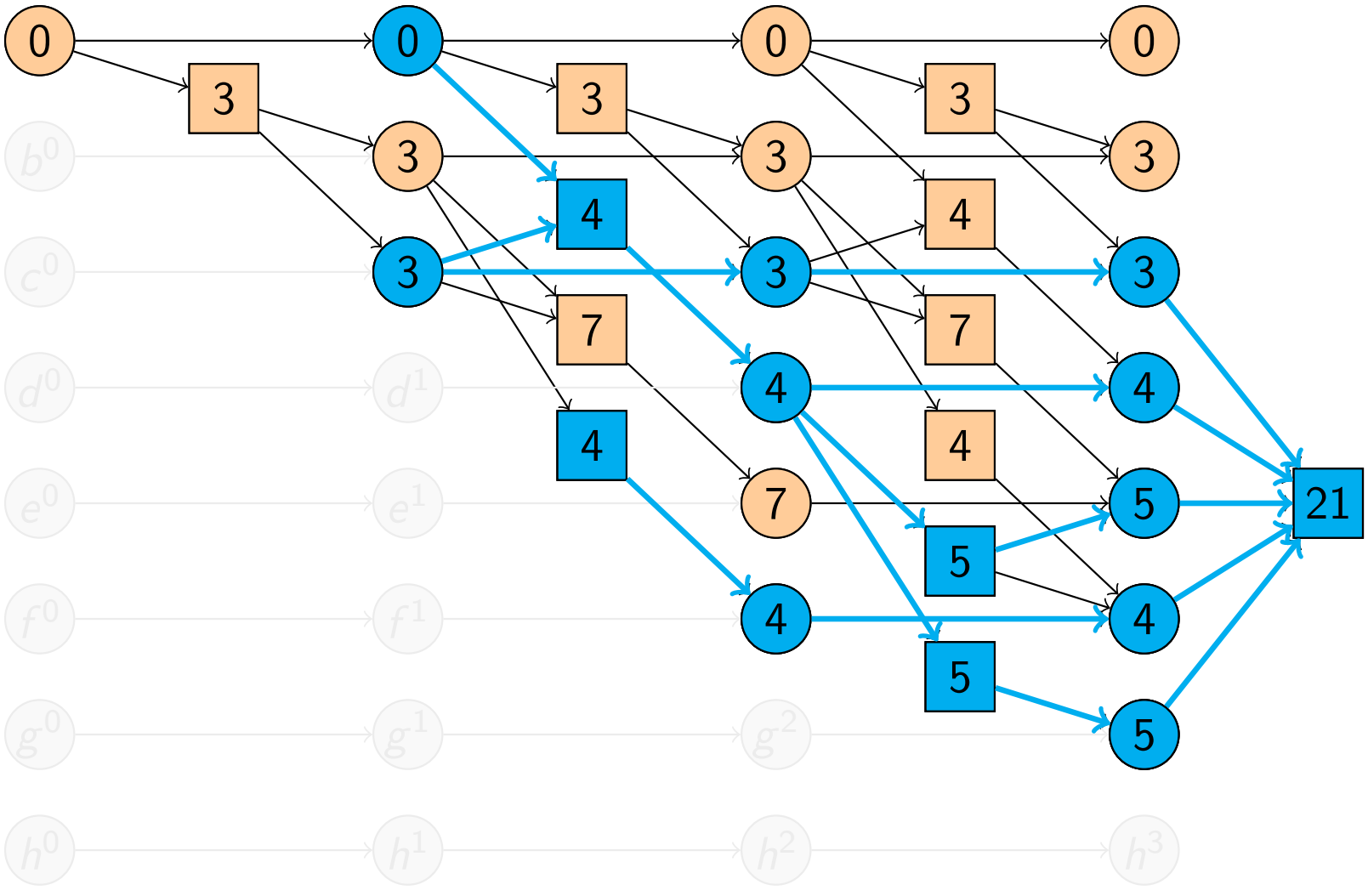
Illustrierendes Beispiel: h^{FF}



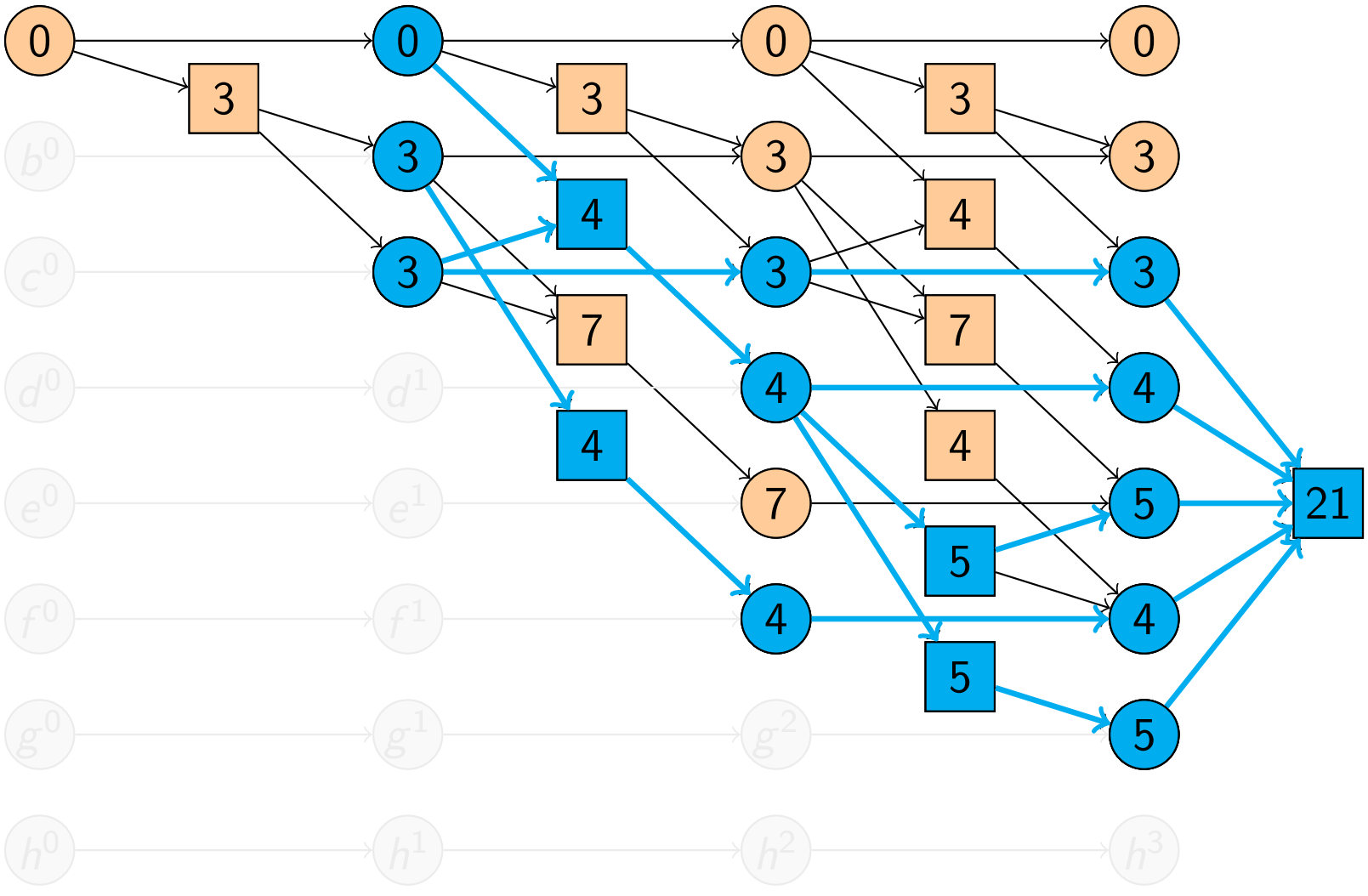
Illustrierendes Beispiel: h^{FF}



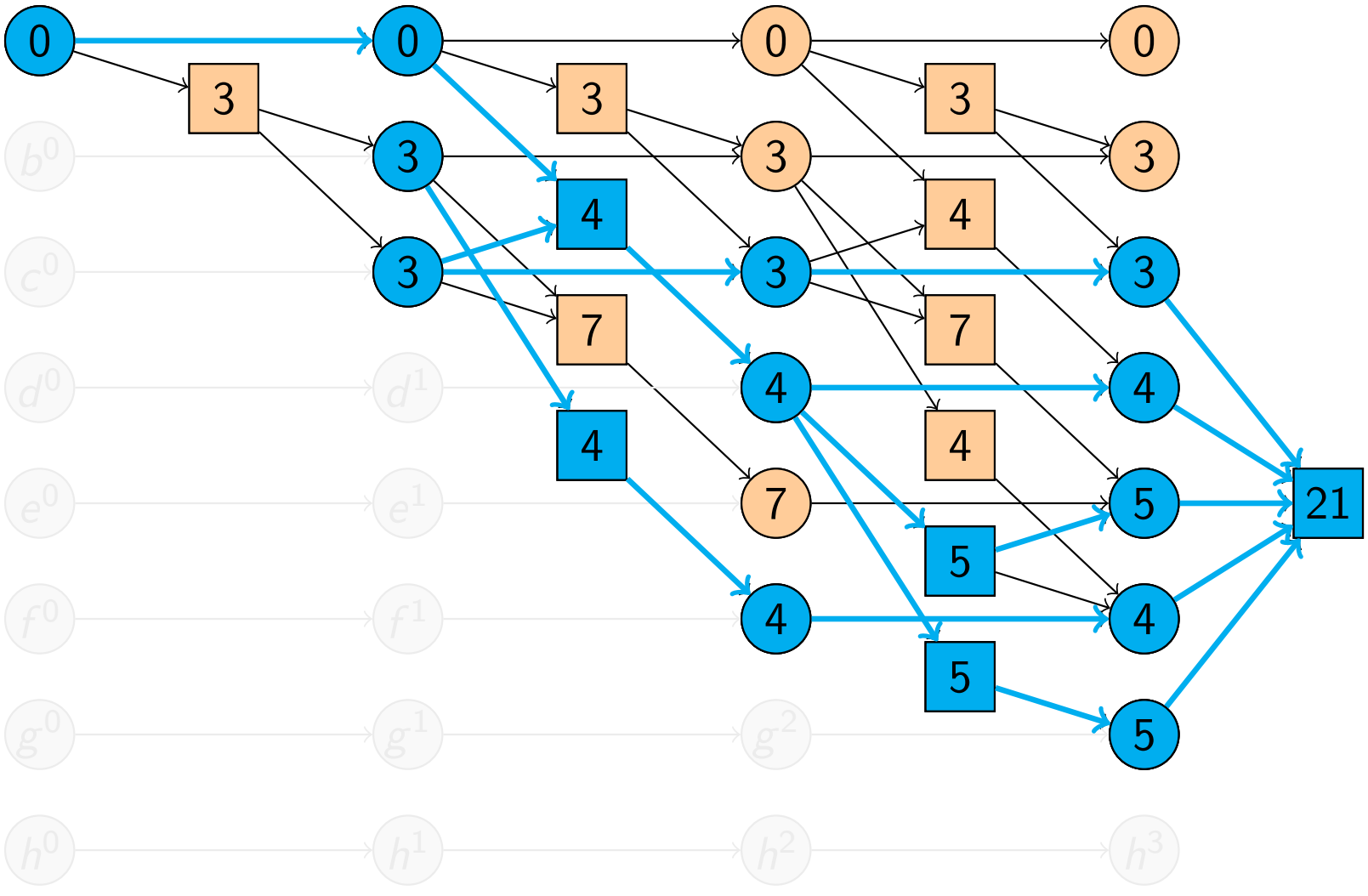
Illustrierendes Beispiel: h^{FF}



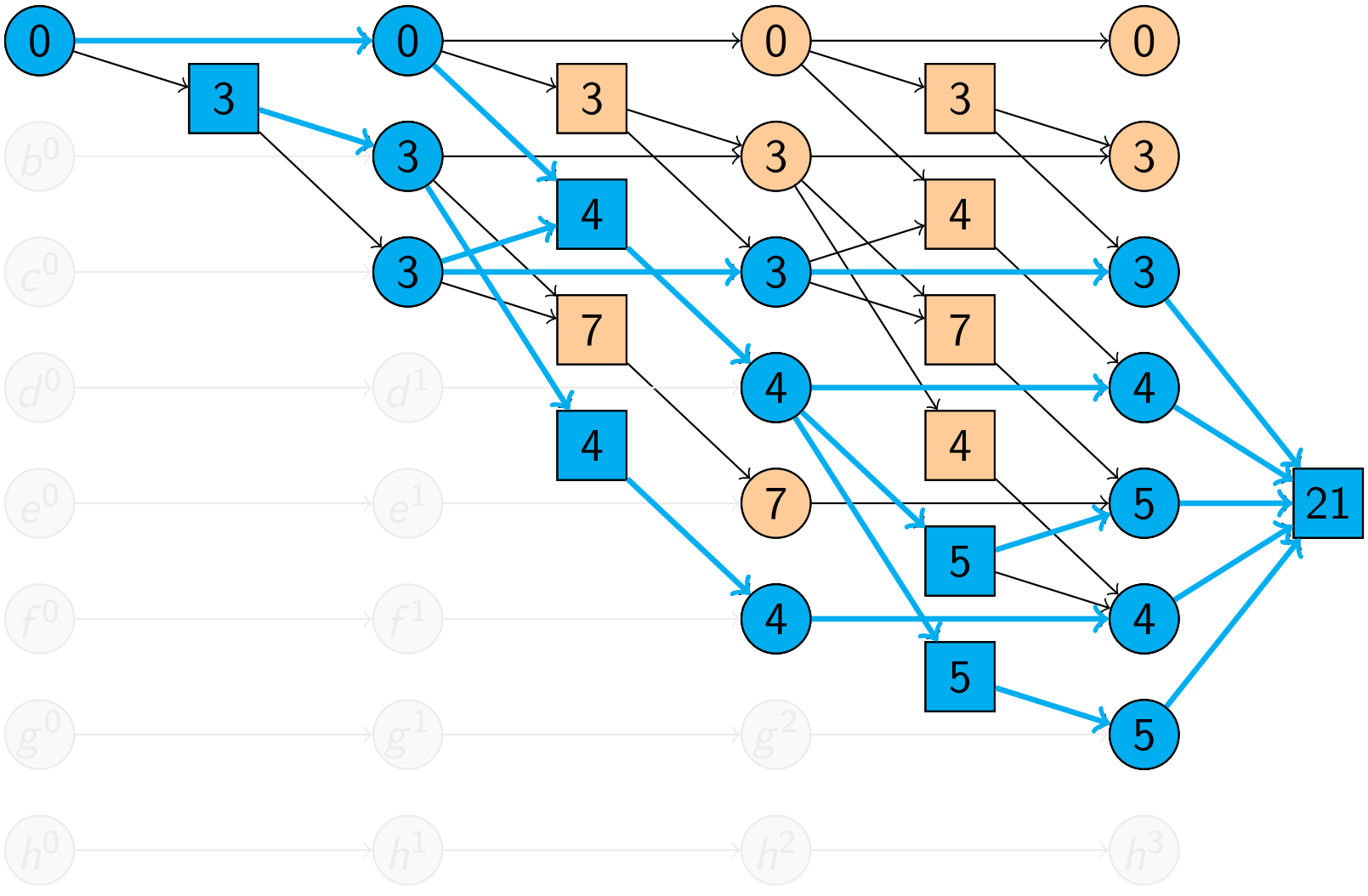
Illustrierendes Beispiel: h^{FF}



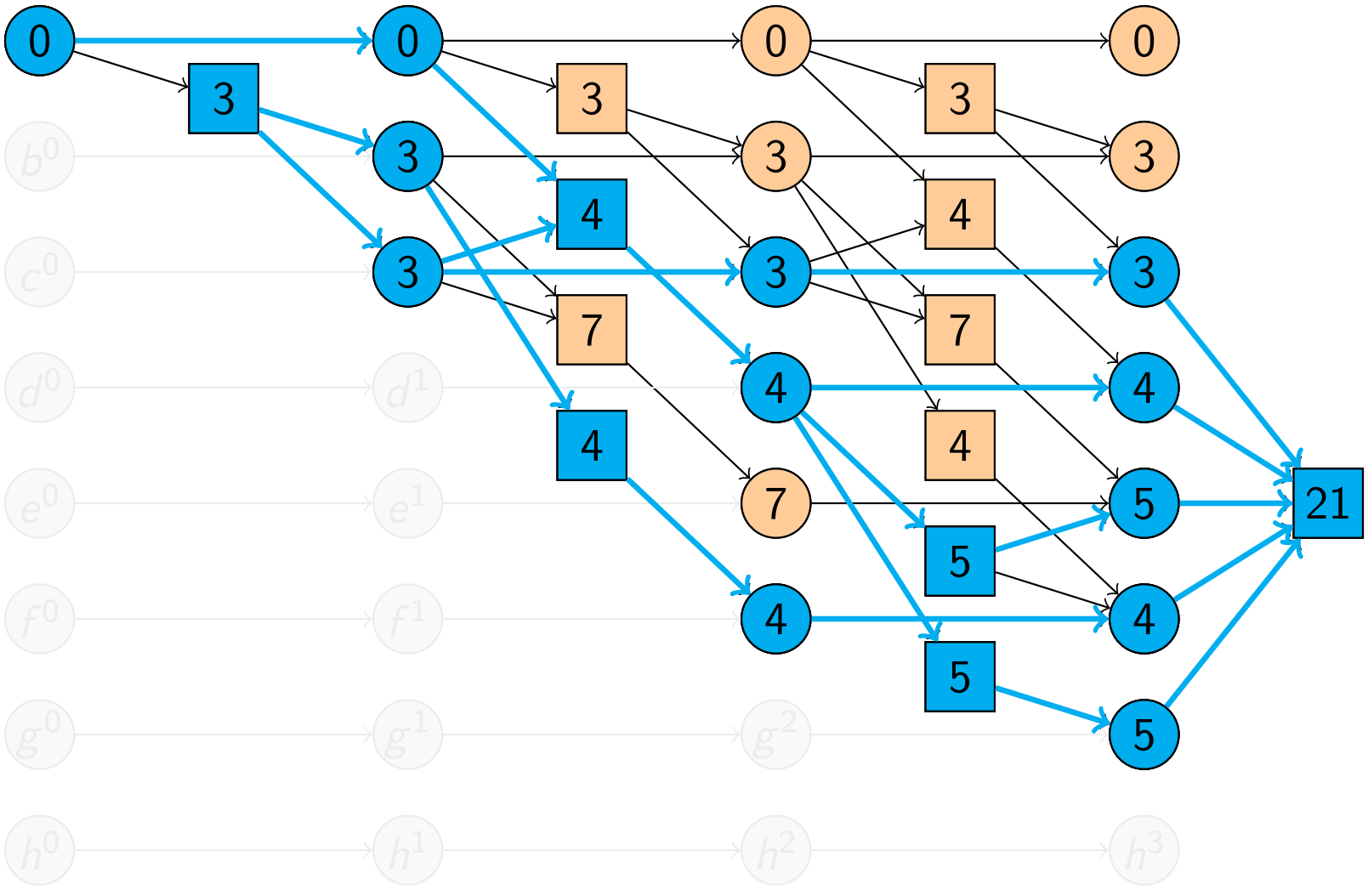
Illustrierendes Beispiel: h^{FF}



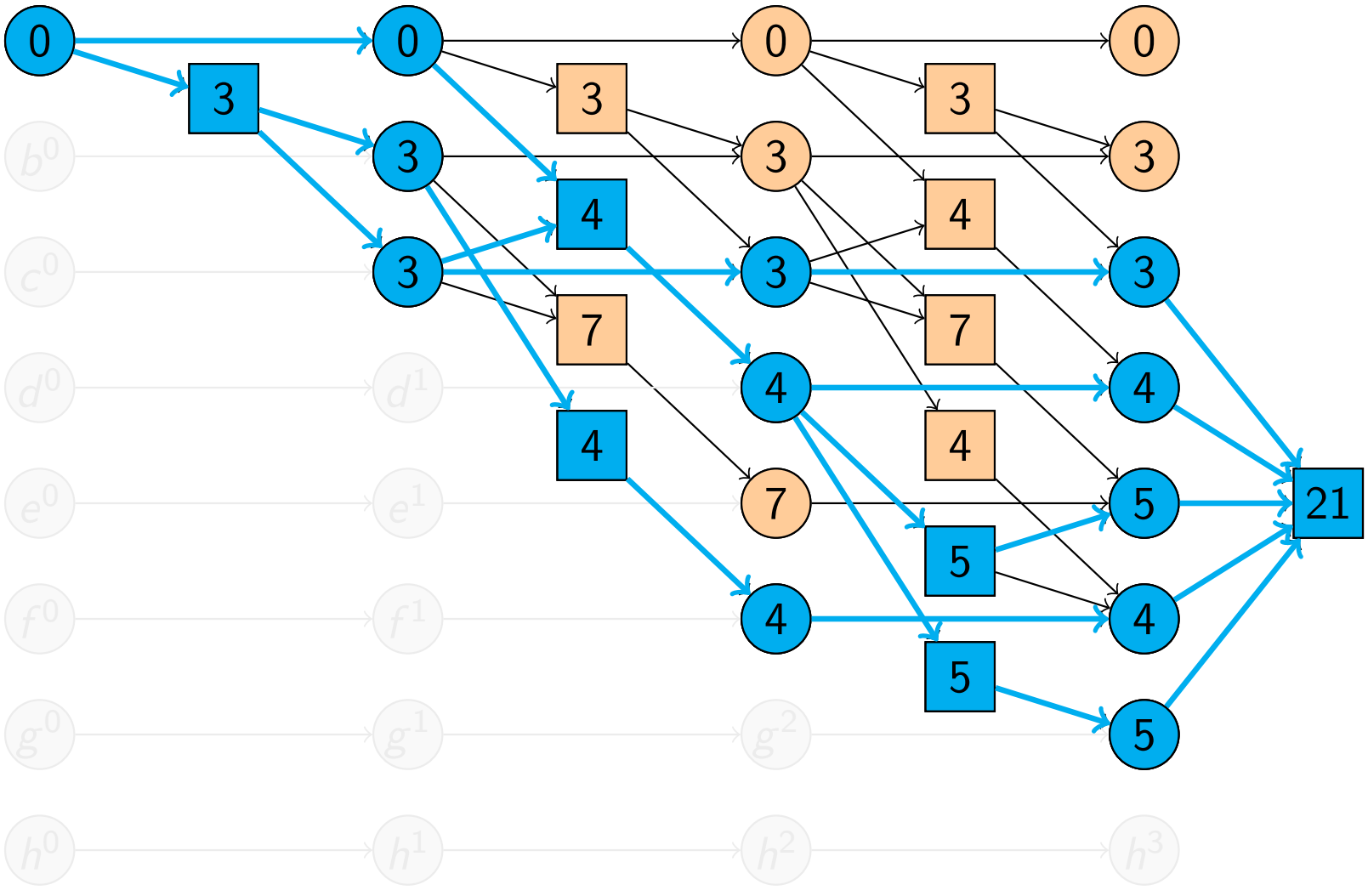
Illustrierendes Beispiel: h^{FF}



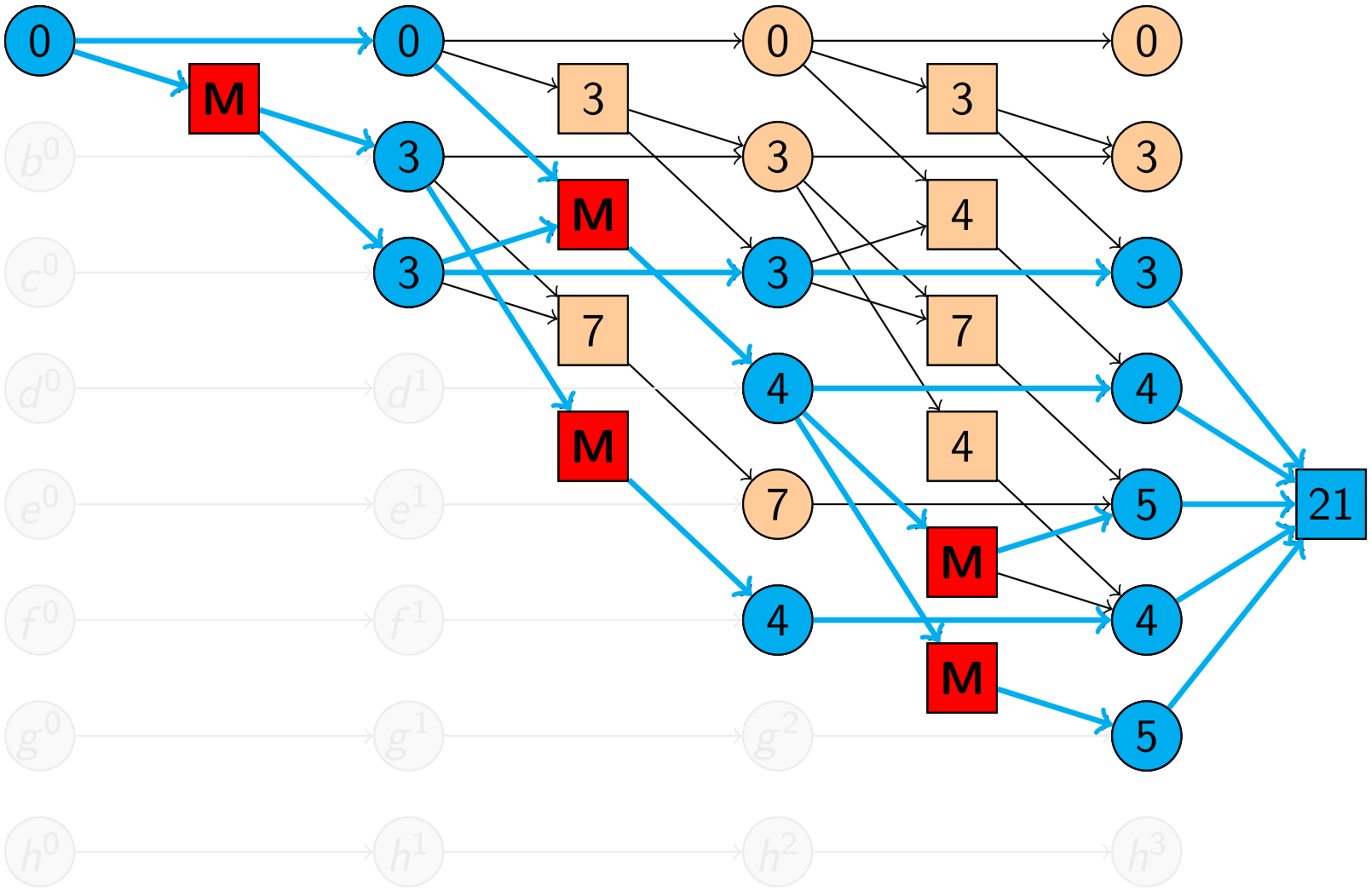
Illustrierendes Beispiel: h^{FF}



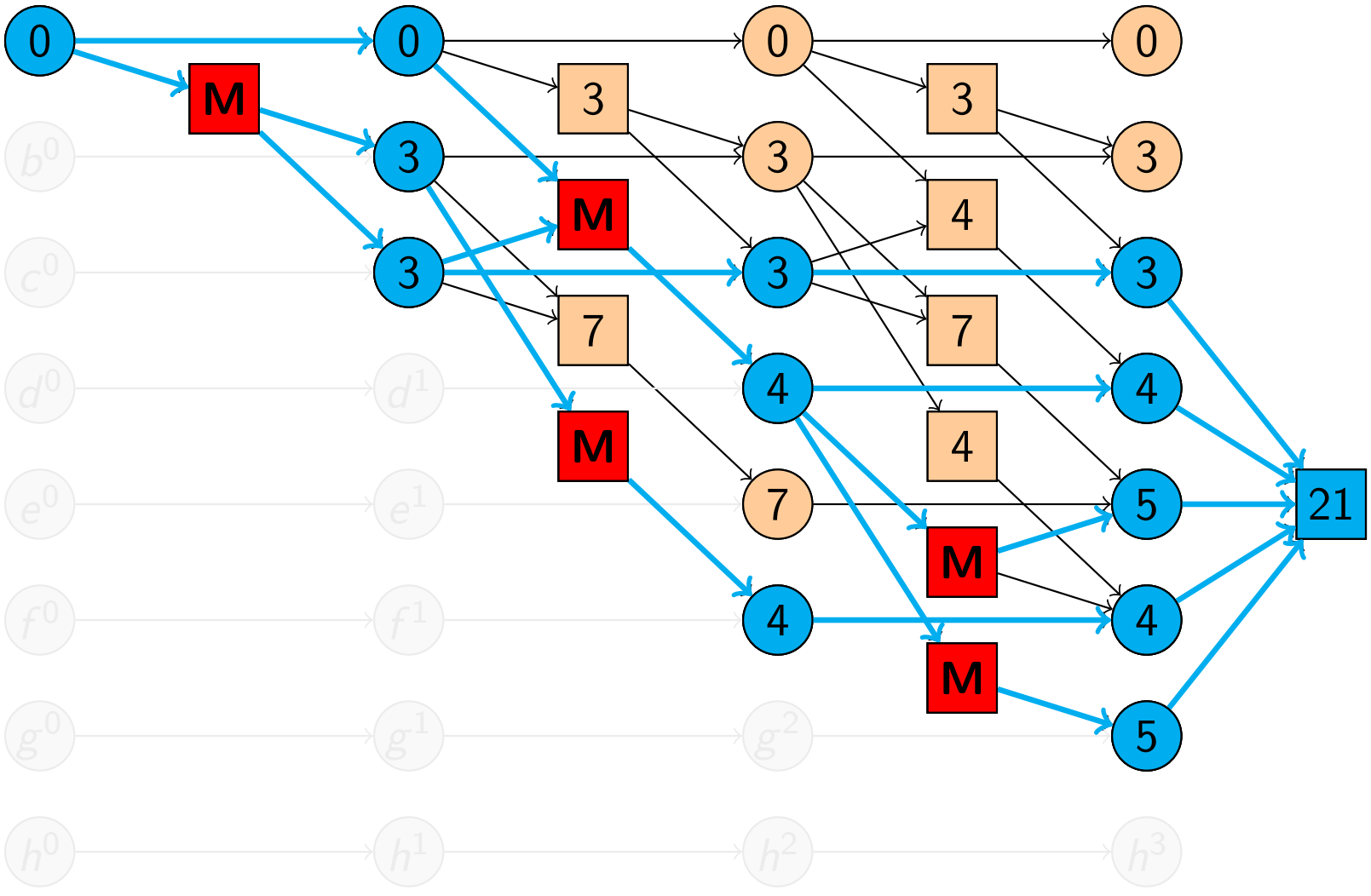
Illustrierendes Beispiel: h^{FF}



Illustrierendes Beispiel: h^{FF}



Illustrierendes Beispiel: h^{FF}



$$h^{\mathrm{FF}}(\{a\}) = 3 + 1 + 1 + 1 + 1 = 7$$

Bemerkungen zur FF-Heuristik

- Wie h^{add} ist h^{FF} sicher und zielerkennend, aber weder zulässig noch konsistent.
- immer mindestens so gute Annäherung an h^+ wie h^{add}
- meistens deutlich besser
- kann in linearer Zeit in der Grösse der Beschreibung der Planungsaufgabe berechnet werden

Bemerkungen zur FF-Heuristik

- Wie h^{add} ist h^{FF} sicher und zielerkennend, aber weder zulässig noch konsistent.
- immer mindestens so gute Annäherung an h^+ wie h^{add}
- meistens deutlich besser
- kann in linearer Zeit in der Grösse der Beschreibung der Planungsaufgabe berechnet werden
- berechneter Heuristik-Wert hängt von Tie-Breaking der Markierungsregeln ab (h^{FF} nicht wohldefiniert)
- eine der erfolgreichsten Planungsheuristiken

Vergleich der Relaxierungsheuristiken

Zusammenhänge zwischen den Relaxierungsheuristiken

Sei s ein Zustand der STRIPS-Planungsaufgabe $\langle V, I, G, A \rangle$.

Dann gilt:

- $h^{\max}(s) \leq h^{+}(s) \leq h^{*}(s)$
- $h^{\max}(s) \leq h^{+}(s) \leq h^{\mathrm{FF}}(s) \leq h^{\mathrm{add}}(s)$
- h^{*} und h^{FF} sind unvergleichbar
- h^{*} und h^{add} sind unvergleichbar

Anmerkungen:

- Für nicht zulässige Heuristiken ist es im Allgemeinen weder gut noch schlecht, grössere Werte zu berechnen als eine andere Heuristik.
- Bei Relaxierungsheuristiken ist das Ziel, möglichst nah an h^{+} zu kommen.