

## Les 8 Proces analyse

Veel processen laten zich door netwerken beschrijven, waarin knopen acties aangeven en opdrachten langs verbindingen tussen de knopen verwerkt worden. Typische elementen van dit soort netwerken zijn:

- **rijen:** een aantal acties worden achter elkaar uitgevoerd,
- **en-splitsingen:** een aantal alternatieven worden parallel toegepast,
- **of-splitsingen:** van een aantal alternatieven wordt er één toegepast,
- **iteratie:** een actie wordt meerdere keren achter elkaar toegepast.

We zullen in deze les kijken hoe we netwerken kunnen analyseren, om bijvoorbeeld de volgende vragen te kunnen beantwoorden:

- (1) Wat is de zwakste schakel in het netwerk, dus welke component van het netwerk moeten we verbeteren om vooruitgang in de algemene prestatie van het netwerk te boeken?
- (2) Hoe kunnen we bepalen wat de gemiddelde tijd is, waarmee een opdracht verwerkt wordt en wat is de tijd waarin 90% van de opdrachten door het netwerk gesluisd wordt?

Om dit soort vraagstukken op te lossen, hebben we natuurlijk zekere informatie over de componenten van het netwerk nodig.

Vaak wordt er voor elke actie in het netwerk een verwerkingstijd aangegeven die het uitvoeren van deze actie in beslag neemt. Dit kan de gemiddelde tijdsduur voor de taak zijn, maar soms ook de tijd waarin een zekere percentage (bijvoorbeeld 95%) van de gevallen voltooid wordt. We zullen zien, dat dit soort informatie voldoende is, om kritieke componenten van een netwerk te identificeren.

Voor vragen over de verdeling van de verwerkingstijden van opdrachten door het netwerk hebben we natuurlijk meer informatie nodig, namelijk kansverdelingen voor de tijdsduur die elke actie in beslag neemt. Deze verdelingen zijn of discrete of continue kansverdelingen.

- (a) **Discrete kansverdeling:** Als we aannemen dat de tijd voor een actie alleen maar veelvouden van een 'eenheidsinterval'  $t_0$  (bijvoorbeeld tijden afgerond op hele seconden) aanneemt, kunnen we een kansverdeling door  $P(T = k \cdot t_0)$  voor  $k = 0, 1, 2, \dots$  aangeven. Hiervoor zullen we kort  $P(k)$  schrijven.

Natuurlijk is het vaak ook interessant, naar de kans te kijken dat de tijdsduur  $T$  van een actie onder een zekere bovengrens ligt. Deze kans is gelijk aan de som over de kansen voor tijden tot en met de grens, we hebben dus  $P(T \leq k \cdot t_0) = \sum_{j=0}^k P(j)$ . Deze kans gaan we met  $F(k)$  afkorten en noemen dit de *discrete verdelingsfunctie*.

Omgekeerd kunnen we uit de discrete verdelingsfunctie  $F(k)$  ook de kansverdeling  $P(k)$  weer makkelijk achterhalen, er geldt namelijk  $P(k) = F(k) - F(k - 1)$ .

- (b) **Continue kansverdeling:** Als we willekeurige positieve tijden toelaten, hebben we een continue kansverdeling nodig. Deze wordt door een dichtheidsfunctie  $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  met  $\int_0^\infty f(t) dt = 1$  aangegeven. De kans  $P(T \leq t)$  dat een actie hoogstens de tijdsduur  $t$  in beslag neemt is dan gegeven door de verdelingsfunctie  $F(t) := \int_0^t f(\tau) d\tau$ , waarbij de integraal de oppervlakte onder de grafiek van  $f(t)$  op het interval  $[0, t]$  aangeeft.

## 8.1 Elementen van netwerken

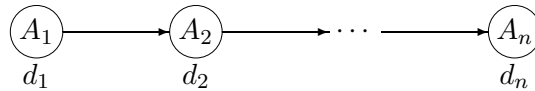
We zullen nu naar de typische componenten van netwerken kijken en hun eigenschappen met betrekking tot verwerkingstijden en tijdsduurverdelingen beschrijven.

### Actie



Een actie  $A$  wordt door een knoop gerepresenteerd. De verwerkingstijd  $d$  (voor *duration*) wordt aangegeven door een waarde die aan de knoop hangt.

### Rij



De verwerkingstijd van een rij is natuurlijk de som

$$d = d_1 + d_2 + \dots + d_n = \sum_{i=1}^n d_i$$

van de verwerkingstijden van de enkele componenten.

Stel nu dat we discrete kansverdelingen voor de tijdsduur van de enkele acties hebben. We hoeven alleen maar naar het geval van twee acties te kijken, voor langere rijen kunnen we het resultaat hiervoor herhaaldelijk toepassen. De kansverdeling voor de actie  $A_1$  noteren we met  $P_1(k) := P_1(T = k)$  en de verdeling voor  $A_2$  met  $P_2(k) := P_2(T = k)$ .

De achter elkaar geschakelde acties  $A_1$  en  $A_2$  nemen  $t$  tijdseenheden in beslag als  $A_1$  een tijd van  $k$  en  $A_2$  een tijd van  $t - k$  duurt. Om de kans hiervoor te berekenen, moeten we over alle mogelijke waarden van  $k$  tussen 0 en  $t$  lopen en krijgen zo

$$P(T = t) = \sum_{k=0}^t P_1(k) \cdot P_2(t - k).$$

Een op deze manier *gevouwen* som van producten speelt in verschillende gebieden van de wiskunde een belangrijke rol (bijvoorbeeld ook bij Fourier transformaties) en heet een (discreet) *convolutieproduct*. Het convolutieproduct wordt meestal met een sterretje genoteerd, dus

$$P(t) = (P_1 * P_2)(t).$$

Let op dat je zo'n convolutieproduct niet met het gewone product  $P_1(t) \cdot P_2(t)$  verwisselt.

Er is een situatie waar we een convolutieproduct toepassen zonder hier verder over na te denken. Als we namelijk twee veeltermen  $f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$  en  $g(x) = b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$  met elkaar vermenigvuldigen, dan is het duidelijk dat we de coëfficiënt van  $x^5$  in het product  $f(x) \cdot g(x)$  vinden als  $a_1b_4 + a_2b_3 + a_3b_2 + a_4b_1$ . We tellen dus de producten van coëfficiënten bij elkaar op die bij machten van  $x$  met som 5 horen. Maar dit is juist een convolutieproduct, namelijk  $\sum_{k=0}^5 a_k b_{5-k}$  (waarbij we  $a_5$  en  $b_5$  als 0 definiëren).

Met behulp van het convolutieproduct kunnen we ook de discrete verdelingsfunctie  $F$  voor de kans op een verwerkingstijd van hoogstens  $t$  uit de verdelingsfuncties  $F_1$  en  $F_2$  voor  $A_1$  en  $A_2$  berekenen. We hebben namelijk

$$\begin{aligned} F(t) &= P(T \leq t) = \sum_{s=0}^t P(T = s) = \sum_{s=0}^t \left( \sum_{k=0}^s P_1(k) \cdot P_2(s-k) \right) \\ &= \sum_{s=0}^t \left( \sum_{k=0}^s P_1(k) \right) \cdot P_2(t-s) = \sum_{s=0}^t F_1(s) \cdot P_2(t-s) = (F_1 * P_2)(t). \end{aligned}$$

Net zo goed kunnen we  $F$  ook door  $F(t) = (P_1 * F_2)(t)$  berekenen.

Op een soortgelijke manier kunnen we de verdeling van de tijdsduren van een rij acties ook voor continue kansverdelingen bepalen. Hierbij nemen de dichtheidsfuncties  $f_1(t)$  en  $f_2(t)$  de plaats van de kansverdelingen  $P_1$  en  $P_2$  in en wordt de som een integraal. De dichtheidsfunctie  $f(t)$  voor de kans dat  $A_1$  en  $A_2$  samen  $t$  seconden in beslag nemen is dan

$$f(t) = \int_0^t f_1(\tau) \cdot f_2(t-\tau) d\tau$$

een dit noemen we het convolutieproduct van de functies  $f_1$  en  $f_2$ , weer genoteerd door  $f(t) = (f_1 * f_2)(t)$ . De verdelingsfunctie voor de gecombineerde verwerkingstijd van de acties  $A_1$  en  $A_2$  is dan

$$F(t) = P(T \leq t) = \int_0^t f(\tau) d\tau = \int_0^t \left( \int_0^\tau f_1(x) \cdot f_2(\tau-x) dx \right) d\tau.$$

**Voorbeeld:** We kijken naar een rij van twee acties  $A_1$  en  $A_2$  met exponentieel verdeelde tijdsduren met parameters  $\lambda_1$  en  $\lambda_2$  (dit betekent dat de verwachtingswaarden voor de verwerkingstijden  $\lambda_1^{-1}$  en  $\lambda_2^{-1}$  zijn). De dichtheidsfunctie voor een exponentiële verdeling met parameter  $\lambda$  is  $\lambda e^{-\lambda t}$ , dus hebben we voor de dichtheidsfunctie van de acties:

$$f(t) = \int_0^t \lambda_1 e^{-\lambda_1 \tau} \cdot \lambda_2 e^{-\lambda_2(t-\tau)} d\tau = \lambda_1 \lambda_2 e^{-\lambda_2 t} \int_0^t e^{(\lambda_2 - \lambda_1)\tau} d\tau.$$

Voor het speciaal geval  $\lambda_1 = \lambda_2 = \lambda$  geeft dit de dichtheidsfunctie

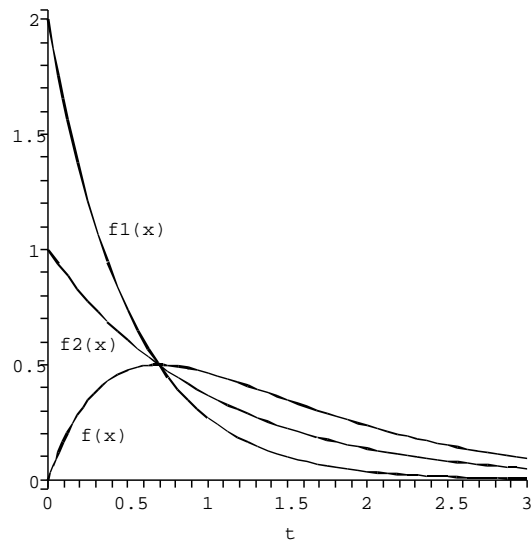
$$f(t) = \lambda^2 t e^{-\lambda t}$$

en in dit geval is de verdelingsfunctie voor de rij van acties

$$F(t) = \int_0^t f(\tau) d\tau = \int_0^t \lambda^2 \tau e^{-\lambda \tau} d\tau = 1 - (1 + \lambda t) e^{-\lambda t}.$$

Voor het algemeen geval  $\lambda_1 \neq \lambda_2$  hebben we

$$f(t) = \lambda_1 \lambda_2 e^{-\lambda_2 t} \frac{1}{\lambda_2 - \lambda_1} (e^{(\lambda_2 - \lambda_1)t} - 1) = \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1} (e^{-\lambda_1 t} - e^{-\lambda_2 t}).$$



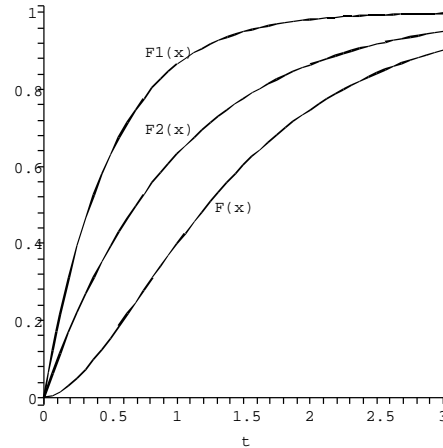
Figuur 9: Dichtheidsfuncties voor twee exponentieel verdeelde acties met parameters  $\lambda_1 = 2$  en  $\lambda_2 = 1$  en de rij van deze acties.

In Figuur 9 zijn de dichtheidsfuncties voor twee exponentiële verdelingen  $f_1(t)$  en  $f_2(t)$  met parameters  $\lambda_1 = 2$  en  $\lambda_2 = 1$  en hun convolutieproduct  $f(t) = (f_1 * f_2)(t)$  te zien. De functies zijn makkelijk te identificeren omdat een exponentiële verdeling met parameter  $\lambda$  in het punt  $t = 0$  de waarde  $\lambda$  heeft.

Om de verdelingsfunctie voor de gecombineerde actie te berekenen, moeten we de dichtheidsfunctie  $f(t)$  nog integreren, we hebben

$$F(t) = \int_0^t \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1} (e^{-\lambda_1 t} - e^{-\lambda_2 t}) dt = 1 - \left( \frac{\lambda_2}{\lambda_2 - \lambda_1} e^{-\lambda_1 t} - \frac{\lambda_1}{\lambda_2 - \lambda_1} e^{-\lambda_2 t} \right).$$

Voor de verdeling  $f(t)$  van de rij van acties laat zich aantonen dat de verwachtingswaarde  $\int_0^\infty f(t) dt = \frac{\lambda_1 + \lambda_2}{\lambda_1 \lambda_2}$  is, dit is juist de som  $\frac{1}{\lambda_1} + \frac{1}{\lambda_2}$  van de verwachtingswaarden van de twee acties. Voor het geval  $\lambda_1 = 2$  en  $\lambda_2 = 1$  is de verwachtingswaarde dus  $\frac{3}{2}$ .

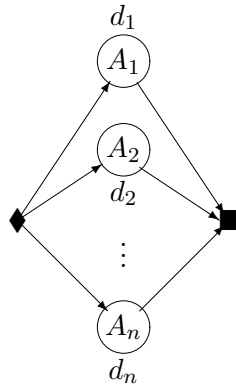


Figuur 10: Verdelingsfuncties voor twee exponentieel verdeelde acties met parameters  $\lambda_1 = 2$  en  $\lambda_2 = 1$  en de rij van deze acties.

In Figuur 10 zijn de verdelingsfuncties  $F_1(t)$ ,  $F_2(t)$  en  $F(t)$  voor de dichtheidsfuncties  $f_1(t)$ ,  $f_2(t)$  en  $f(t)$  van de acties van boven te zien. Het is duidelijk dat de verdelingsfunctie voor de verdeling met de kleinste verwachtingswaarde het hoogste ligt, daarom zijn de functies van boven naar beneden  $F_1(T)$ ,  $F_2(T)$ ,  $F(T)$ .

### En-splitsing

Een en-splitsing ligt tussen een ingevulde ruit  $\blacklozen$  en een ingevulde vierkant  $\blacksquare$ .



Als we een en-splitsing met de alternatieve acties  $A_1, A_2, \dots, A_n$  hebben, die alle parallel uitgevoerd moeten worden, is de verwerkingstijd  $d$  natuurlijk het maximum van de verwerkingstijden van de enkele componenten, dus we hebben

$$d = \max\{d_1, d_2, \dots, d_n\}.$$

Om de kansverdeling voor de tijdsduur van de splitsing te vinden, gebruiken we de verdelingsfuncties  $F_i$  van de enkele acties  $A_i$ . Hierbij maakt het niets uit of we met discrete of continue kansverdelingen te maken hebben. De en-splitsing

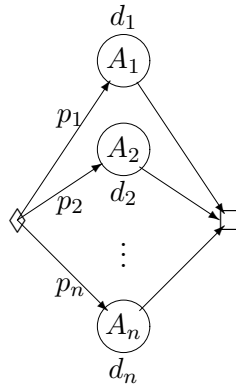
verwerkt een opdracht in een tijd  $T \leq t$  als elke actie  $A_i$  de opdracht in een tijd  $T \leq t$  verwerkt. Omdat we aannemen dat de tijden voor de verschillende acties onafhankelijk van elkaar zijn, is de kans dat alle acties  $A_i$  hoogstens een duur van  $t$  hebben juist het product van de kansen voor de enkele acties, daarom hebben we voor de verdelingsfunctie  $F(t)$  van de en-splitsing:

$$F(t) = F_1(t) \cdot F_2(t) \cdot \dots \cdot F_n(t) = \prod_{i=1}^n F_i(t).$$

Als de kansverdelingen discreet zijn, krijgen we uit de verdelingsfunctie de kans  $P(k) = P(T = k \cdot t_0)$  makkelijk door  $P(k) = F(k) - F(k - 1)$ . Voor continue kansverdelingen moeten we de verdelingsfunctie afleiden en krijgen de dichtheidsfunctie als  $f(t) = F'(t)$ .

### Of-splitsing

Een of-splitsing ligt tussen een niet ingevulde ruit  $\diamond$  en een niet ingevulde vierkant  $\square$ .



Een of-splitsing bestaat uit alternatieve acties, waarvan slechts één uitgevoerd hoeft te worden. Als aanvullende informatie moeten we hierbij nog aangeven, met welke kans  $p_i$  de enkele acties ingegaan worden.

Voor een *worst-case* analyse zouden we ook hier het maximum van de verwerkingstijden van de enkele acties moeten nemen. Maar als we willen bepalen, wat voor een tijd we typisch kunnen verwachten, is het voor de hand liggend de verwachtingswaarde van de verwerkingstijden te berekenen, dus we krijgen

$$d = p_1 d_1 + p_2 d_2 + \dots + p_n d_n = \sum_{i=1}^n p_i d_i.$$

Net als bij de en-splitsing gebruiken we de verdelingsfuncties van de enkele acties om de verdelingsfunctie voor de hele splitsing te vinden. De kans, dat de splitsing een opdracht in een tijd  $T \leq t$  verwerkt is de gewogen som van de kansen dat een enkele actie de opdracht in een tijd  $T \leq t$  verwerkt. De gewichten zijn hierbij de kansen  $p_i$  waarmee de actie  $A_i$  ingegaan wordt. We hebben dus (weer voor discrete en continue kansverdelingen):

$$F(t) = p_1 F_1(t) + p_2 F_2(t) + \dots + p_n F_n(t) = \sum_{i=1}^n p_i F_i(t).$$

Omdat we de verdelingsfunctie als gewogen som krijgen, kunnen we er ook meteen de kansverdeling of dichtheidsfunctie van afleiden. In het discrete geval geldt

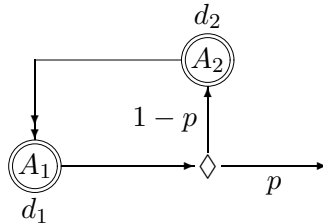
$$P(k) = P(T = k) = p_1P_1(k) + p_2P_2(k) + \dots + p_nP_n(k) = \sum_{i=1}^n p_iP_i(k)$$

en in het continue geval

$$f(t) = P(T \leq t)' = p_1f_1(t) + p_2f_2(t) + \dots + p_nf_n(t) = \sum_{i=1}^n p_if_i(t).$$

Als we aannemen dat de bij actie  $A_i$  aangegeven verwerkingstijd  $d_i$  de verwachtingswaarde van de kansverdeling voor deze actie is, volgt hieruit dat de gemiddelde kans  $d = \sum_{i=1}^n p_id_i$  juist de verwachtingswaarde voor de verwerkingstijd door de hele of-splitsing is.

### Iteratie



In een iteratie hebben we een actie  $A_1$ , waarna een beslissing wordt genomen of we door gaan (dit gebeurt met kans  $p$ ) of via een actie  $A_2$  terug naar  $A_1$  gaan (dit gebeurt met kans  $1 - p$ ). Beide acties  $A_1$  en  $A_2$  kunnen in principe leeg zijn en een duur van 0 hebben.

In principe kunnen we een iteratie zien als een speciale of-splitsing met oneindig veel alternatieven, want we kunnen de lus willekeurig vaak doorlopen voor dat we er uit komen. De alternatieven en hun verwerkingstijden zijn dus:

0 iteraties: actie  $A_1$ , tijd  $d_1$ , kans  $p$

1 iteratie: acties  $A_1, A_2, A_1$ , tijd  $2d_1 + d_2$ , kans  $(1 - p)p$

2 iteraties: acties  $A_1, A_2, A_1, A_2, A_1$ , tijd  $3d_1 + 2d_2$ , kans  $(1 - p)^2p$

⋮

$k$  iteraties: acties  $A_1, A_2, A_1, \dots, A_2, A_1$ , tijd  $(k + 1)d_1 + kd_2$ , kans  $(1 - p)^k p$

Als we  $k$  keer itereren voeren we dus  $k + 1$  keer de actie  $A_1$  en  $k$  keer de actie  $A_2$  uit en hebben hiervoor de tijd  $(k + 1)d_1 + kd_2$  nodig en dit gebeurt met kans  $(1 - p)^k p$ .

Zo als bij de of-splitsing bepalen we de verwerkingstijd als verwachtingswaarde voor de verwerkingstijden van de enkele alternatieven, maar deze keer is dit een som over oneindig veel alternatieven, omdat we de lus willekeurig vaak kunnen doorlopen. Om dit uit te werken, hebben we als hulpmiddel de *meetkundige reeks* nodig.

De meetkundige reeks is de oneindige som  $1 + x + x^2 + \dots = \sum_{k=0}^{\infty} x^k$  en er geldt dat

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$$

als  $|x| < 1$  is. (Als  $|x| > 1$  is, bestaat de som niet, omdat de waarden van  $x^k$  steeds groter worden.) Deze gelijkheid ziet men in door na te gaan dat  $(1-x)(\sum_{k=0}^n x^k) = 1 - x^{n+1}$  en dus  $\sum_{k=0}^n x^k = \frac{1-x^{n+1}}{1-x}$ . Als  $|x| < 1$  gaat  $x^n \rightarrow 0$  voor  $n \rightarrow \infty$  en dus gaat  $\sum_{k=0}^{\infty} x^k \rightarrow \frac{1}{1-x}$ .

Als een eerste toepassing van de meetkundige reeks kunnen we nagaan dat de kansen voor het doorlopen van 0, 1, 2, enz. iteraties inderdaad som 1 hebben. De kans voor  $k$  iteraties is  $(1-p)^k p$  en we hebben

$$\sum_{k=0}^{\infty} (1-p)^k p = \left( \sum_{k=0}^{\infty} (1-p)^k \right) \cdot p = \frac{1}{1-(1-p)} \cdot p = \frac{1}{p} \cdot p = 1.$$

Als we de relatie  $\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$  op beide zijden naar  $x$  afleiden (waarbij we de som aan de linkerkant termgewijs afleiden), krijgen we  $\sum_{k=0}^{\infty} kx^{k-1} = (\sum_{k=0}^{\infty} x^k)' = (\frac{1}{1-x})' = \frac{1}{(1-x)^2}$ . Als we hier voor  $x$  weer  $1-p$  invullen, hebben we

$$\sum_{k=0}^{\infty} k(1-p)^{k-1} = \frac{1}{(1-(1-p))^2} = \frac{1}{p^2}.$$

Voor de verwachtingswaarde van de verwerkingstijd van de iteratie volgt hiermee:

$$\begin{aligned} d &= \sum_{k=0}^{\infty} (1-p)^k p ((k+1)d_1 + kd_2) \\ &= pd_1 \cdot \sum_{k=0}^{\infty} (1-p)^k (k+1) + pd_2 \cdot \sum_{k=0}^{\infty} (1-p)^k k \\ &= pd_1 \cdot \sum_{k=0}^{\infty} (1-p)^k (k+1) + pd_2(1-p) \cdot \sum_{k=0}^{\infty} (1-p)^{k-1} k \\ &= pd_1 \cdot \frac{1}{p^2} + pd_2(1-p) \cdot \frac{1}{p^2} = \frac{d_1}{p} + \frac{d_2(1-p)}{p} = \frac{d_1 + d_2}{p} - d_2 \end{aligned}$$

Door de interpretatie van de iteratie als of-splitsing kunnen we in principe ook de resultaten van boven over de kansverdelingen voor de tijdsduur toepassen. We hebben het echter hierbij met een oneindige som te maken, waarin steeds langere convolutieproducten voorkomen. In de praktijk wordt daarom een maximum voor het aantal doorlopen van de iteratie gekozen en de som daar afgebroken. Dit is een redelijke aanpak, omdat  $(1-p)^k$  met groeiende  $k$  snel klein wordt en de verdere termen daarom bijna geen rol meer spelen.

**Merk op:** Bij het afbreken van een iteratie moeten we opletten, dat we de kansmassa van de *vergeten* opties niet kwijt raken. Als we bijvoorbeeld na



één rondje van de iteratie afbreken, hebben we een of-splitsing met de optie  $A_1$  met kans  $p$  en de optie  $A_1, A_2, A_1$  met kans  $(1-p)p$ . Maar deze twee kansen bij elkaar opgeteld geven niet 1, maar slechts  $1 - (1-p)^2$ , we zijn namelijk de iteraties met 2 of meer lusjes vergeten. De eenvoudigste (en ook redelijkste) manier om dit op te lossen, is, de kansen voor de opties met 2 of meer lusjes bij de optie met 1 iteratie op te tellen. Dit kunnen we zo interpreteren, dat we na 1 iteratie gedwongen zijn om de iteratie te verlaten.

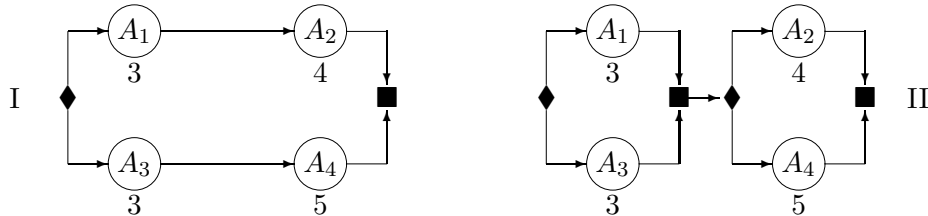
Als we met deze aanpak bij  $k = 2$  afbreken, krijgen we (in het continue geval) de dichtheidsfunctie:

$$f(t) = pf_1(t) + p(1-p)(f_1 * f_2 * f_1)(t) + (1-p)^2(f_1 * f_2 * f_1 * f_2 * f_1)(t).$$

Merk op dat we bij  $f_1 * f_2 * f_1 * f_2 * f_1$  eigenlijk de kans  $p(1-p)^2$  zouden hebben, maar deze is nu tot  $(1-p)^2$  verhoogd.

**Twee voorbeelden**

We gaan de behandelde principes nu eens op de twee volgende eenvoudige netwerken toepassen:



Als we de verwerkingstijden voor de netwerken bepalen, krijgen we  $d = \max(3 + 4, 3 + 5) = 8$  voor netwerk I en  $d = \max(3, 3) + \max(4, 5) = 3 + 5 = 8$  voor netwerk II.

Nu veronderstellen we dat de aangegeven verwerkingstijden verwachtingswaarden zijn en dat de kansverdeling voor de actie  $A_i$  met verwachtingswaarde  $d_i$  gegeven is door

$$P_i(d_i - 1) = 0.25, \quad P_i(d_i) = 0.5, \quad P_i(d_i + 1) = 0.25.$$

Dit betekent dat de helft van de acties in de aangegeven tijd uitgevoerd worden, dat een kwart om een tijdseenheid sneller en een kwart om een tijdseenheid langzamer uitgevoerd wordt. Voor deze kansverdelingen gaan we nu de verdelingen van de verwerkingstijden voor de twee netwerken bepalen. We beginnen met netwerk I.

Voor de kansverdeling  $P_{12}(t)$  van de rij  $A_1 - A_2$  van acties krijgen we met behulp van het convolutieproduct:

$$\begin{aligned} P_{12}(5) &= 0.25 \cdot 0.25 = 0.0625 \\ P_{12}(6) &= 0.25 \cdot 0.5 + 0.5 \cdot 0.25 = 0.25 \\ P_{12}(7) &= 0.25 \cdot 0.25 + 0.5 \cdot 0.5 + 0.25 \cdot 0.25 = 0.375 \\ P_{12}(8) &= 0.5 \cdot 0.25 + 0.25 \cdot 0.5 = 0.25 \\ P_{12}(9) &= 0.25 \cdot 0.25 = 0.0625. \end{aligned}$$

De verdelingsfunctie  $F_{12}(t)$  voor deze rij van acties is dus

$$\begin{aligned} F_{12}(4) &= 0 \\ F_{12}(5) &= F_{12}(4) + P_{12}(5) = 0 + 0.0625 = 0.0625 \\ F_{12}(6) &= F_{12}(5) + P_{12}(6) = 0.0625 + 0.25 = 0.3125 \\ F_{12}(7) &= F_{12}(6) + P_{12}(7) = 0.3125 + 0.375 = 0.6875 \\ F_{12}(8) &= F_{12}(7) + P_{12}(8) = 0.6875 + 0.25 = 0.9375 \\ F_{12}(9) &= F_{12}(8) + P_{12}(9) = 0.9375 + 0.0625 = 1. \end{aligned}$$

Op dezelfde manier krijgen we voor de rij  $A_3 - A_4$  van acties de kansverdeling  $P_{34}(t)$  en verdelingsfunctie  $F_{34}(t)$  met

$$\begin{aligned} P_{34}(5) &= 0, & F_{34}(5) &= 0 \\ P_{34}(6) &= 0.0625, & F_{34}(6) &= 0.0625 \\ P_{34}(7) &= 0.25, & F_{34}(7) &= 0.3125 \\ P_{34}(8) &= 0.375, & F_{34}(8) &= 0.6875 \\ P_{34}(9) &= 0.25, & F_{34}(9) &= 0.9375 \\ P_{34}(10) &= 0.0625, & F_{34}(10) &= 1. \end{aligned}$$

Omdat we het met een en-splitsing te maken hebben, krijgen we de verdelingsfunctie  $F(t)$  voor het hele netwerk als product van de verdelingsfuncties  $F_{12}(t)$  en  $F_{34}(t)$ , dus er geldt  $F(t) = F_{12}(t) \cdot F_{34}(t)$ . Dit geeft de verdelingsfunctie

$$\begin{aligned} F(5) &= 0.0625 \cdot 0 = 0 \\ F(6) &= 0.3125 \cdot 0.0625 = 0.0195 \\ F(7) &= 0.6875 \cdot 0.3125 = 0.2148 \\ F(8) &= 0.9375 \cdot 0.6875 = 0.6445 \\ F(9) &= 1 \cdot 0.9375 = 0.9375 \\ F(10) &= 1 \cdot 1 = 1 \end{aligned}$$

Door  $P(t) = F(t) - F(t-1)$  krijgen we hieruit de kansverdeling  $P(t)$  voor het netwerk, dus

$$P(6) = 0.0195, P(7) = 0.1953, P(8) = 0.4297, P(9) = 0.2930, P(10) = 0.0625.$$

We kunnen nu ook de verwachtingswaarde van de verwerkingstijd door netwerk I bepalen, dit is gewoon de verwachtingswaarde van de kansverdeling  $P$ , dus

$$\sum_{t=6}^{10} t \cdot P(t) \approx 8.18.$$

Het is geen verrassing dat dit hoger ligt dan de verwachtingswaarde voor beide deelpaden van de en-splitsing, want de rij  $A_3 - A_4$  heeft inderdaad verwachtingswaarde 8, maar de gevallen waar de rij  $A_1 - A_2$  een langere verwerkingstijd dan 8 hebben veroorzaken een hogere verwachtingswaarde voor het netwerk.

Het netwerk II kunnen we op een soortgelijke manier analyseren. Voor de verdelingsfuncties  $F_i(t)$  van de enkele acties  $A_i$  geldt

$$F_i(d_i - 2) = 0, \quad F_i(d_i - 1) = 0.25, \quad F_i(d_i) = 0.75, \quad F_i(d_i + 1) = 1.$$

Hieruit krijgen we voor de en-splitsing met acties  $A_1$  en  $A_3$  de verdelingsfunctie  $F_{13}(t) = F_1(t) \cdot F_3(t)$  als

$$F_{13}(2) = 0.25 \cdot 0.25 = 0.0625, \quad F_{13}(3) = 0.75 \cdot 0.75 = 0.5625, \quad F_{13}(4) = 1$$

en dus de kansverdeling  $P_{13}(t)$  met

$$P_{13}(2) = 0.0625, \quad P_{13}(3) = 0.5625 - 0.0625 = 0.5, \quad P_{13}(4) = 1 - 0.5625 = 0.4375.$$

Voor de en-splitsing met acties  $A_2$  en  $A_4$  krijgen we analoog de verdelingsfunctie  $F_{24}(t) = F_2(t) \cdot F_4(t)$  als

$$F_{24}(3) = 0.25 \cdot 0 = 0, \quad F_{24}(4) = 0.75 \cdot 0.25 = 0.1875, \\ F_{24}(5) = 1 \cdot 0.75 = 0.75, \quad F_{24}(6) = 1$$

en de kansverdeling  $P_{24}(t)$  met

$$P_{24}(4) = 0.1875, \quad P_{24}(5) = 0.75 - 0.1875 = 0.5625, \quad P_{24}(6) = 1 - 0.75 = 0.25.$$

De kansverdeling  $P(t)$  voor de verwerkingstijd door het netwerk II krijgen we nu met behulp van het convolutieproduct van  $P_{13}(t)$  en  $P_{24}(t)$ , dit geeft:

$$P(6) = P_{13}(2) \cdot P_{24}(4) = 0.0625 \cdot 0.1875 = 0.0117 \\ P(7) = P_{13}(2) \cdot P_{24}(5) + P_{13}(3) \cdot P_{24}(4) \\ = 0.0625 \cdot 0.5625 + 0.5 \cdot 0.1875 = 0.1289 \\ P(8) = P_{13}(2) \cdot P_{24}(6) + P_{13}(3) \cdot P_{24}(5) + P_{13}(4) \cdot P_{24}(4) \\ = 0.0625 \cdot 0.25 + 0.5 \cdot 0.5625 + 0.4375 \cdot 0.1875 = 0.3789 \\ P(9) = P_{13}(3) \cdot P_{24}(6) + P_{13}(4) \cdot P_{24}(5) \\ = 0.5 \cdot 0.25 + 0.4375 \cdot 0.5625 = 0.3711 \\ P(10) = P_{13}(4) \cdot P_{24}(6) = 0.4375 \cdot 0.25 = 0.1094$$

In dit geval vinden we als verwachtingswaarde voor de verwerkingstijd door netwerk II

$$\sum_{t=6}^{10} t \cdot P(t) \approx 8.44.$$

Het is ook hier geen verrassing dat de verwerking door netwerk II langer duurt, want we moeten in ieder geval met de acties  $A_2$  en  $A_4$  wachten tot dat  $A_1$  en  $A_3$  beide klaar zijn.

## 8.2 Kritieke pad analyse

Als we in een netwerk de zwakste schakel willen vinden, moeten we het pad door het netwerk vinden dat de grootste verwerkingstijd heeft. Dit noemen we een *kritiek pad*. We hebben gezien hoe we voor de verschillende elementen van een netwerk de verwerkingstijden moeten combineren, en dat moeten we bij een netwerk stapsgewijs toepassen. Voor een rij van acties is er natuurlijk geen keuze, maar bij een splitsing hoeven we in principe alleen maar naar het alternatief met de maximale verwerkingstijd te kijken.

Bij een en-splitsing is dit helder, maar bij een of-splitsing leidt dit ertoe dat we alleen maar het ergste geval bekijken, dit noemen we een *worst-case* analyse.

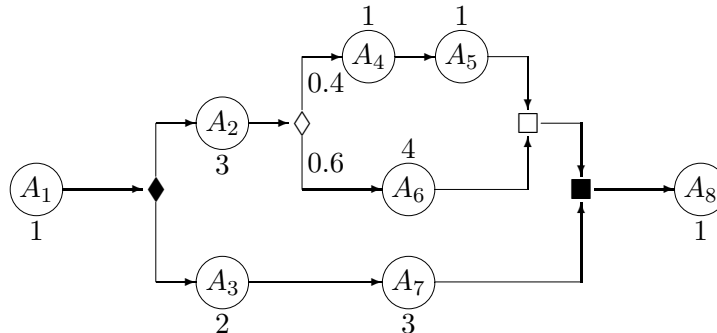
Het kan natuurlijk zo zijn dat de ergste optie van een of-splitsing slechts met een kleine kans ingegaan wordt, daarom kijkt men bij een *gewone analyse* naar alle alternatieven van een of-splitsing. Als verwerkingstijd van de of-splitsing neemt men dan de verwachtingswaarde van de verwerkingstijden van de alternatieven, maar alle acties van de of-splitsing komen in het kritieke pad terecht.

Bij een iteratie krijgen we in een worst-case analyse een oneindige verwerkingstijd, omdat we de lus willekeurig vaak kunnen doorlopen. Daarom behandelen we een iteratie altijd zo als een of-splitsing in de gewone analyse met de verwachtingswaarde van de verwerkingstijd als looptijd.

We krijgen zo de volgende strategie voor het vinden van een kritiek pad:

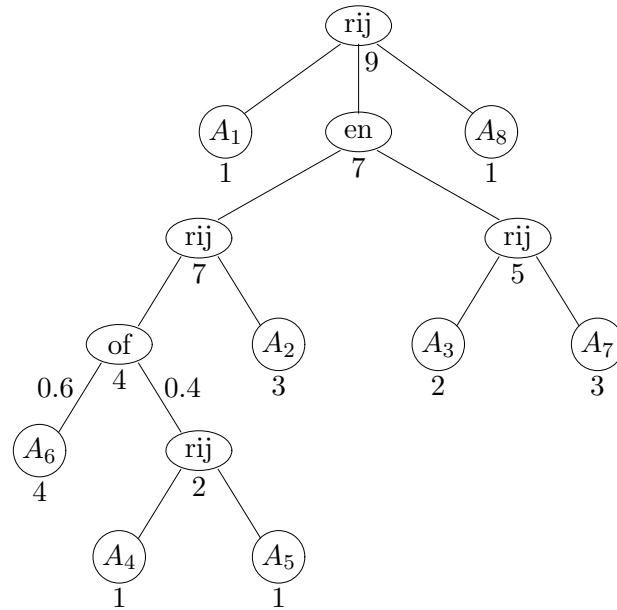
- En-splitsing: alleen maar alternatief met maximale verwerkingstijd in het kritieke pad.
- Of-splitsing:
  - worst-case analyse: alleen maar alternatief met maximale verwerkingstijd in het kritieke pad.
  - gewone analyse: alle alternatieven in het kritieke pad; verwerkingstijd is de verwachtingswaarde van de verwerkingstijden van de alternatieven.
- Iteratie: alle acties van de iteratie in het kritieke pad; verwerkingstijd is de verwachtingswaarde van de looptijd van de iteratie.

We kijken naar het volgende voorbeeld van een netwerk:



Een handige manier om zo'n netwerk te representeren is een *boom* met rijen, en-splitsingen en of-splitsingen als knopen en de acties als bladeren. De takken

vanuit een knoop zijn bij een rij de componenten van de rij (dit kunnen acties, maar ook splitsingen zijn) en bij splitsingen de alternatieven van de splitsing. In zo'n boom laten zich de verwerkingstijden makkelijk vanuit de bladeren tot de wortel bepalen. Voor het gegeven netwerk is dit in Figuur 11 te zien.



Figuur 11: Boom structuur voor een netwerk

De verwerkingstijden zijn van beneden naar boven bepaald, bij een rij is de verwerkingstijd natuurlijk de som van de tijden voor de takken, bij een en- of of-splitsing het maximum van de alternatieven (worst-case analyse).

Het is duidelijk dat de acties  $A_1$  en  $A_8$  in ieder pad door het netwerk bevat zijn. Het lagere pad in de en-splitsing met de acties  $A_3$  en  $A_7$  heeft een verwerkingstijd van  $2 + 3 = 5$ . Voor een pad door  $A_2$  moeten we nog naar de of-splitsing met  $A_4$ ,  $A_5$  en  $A_6$  kijken. Als het om een worst-case analyse gaat, hebben we hier alleen maar het alternatief met de maximale tijd nodig, dat is in dit geval  $A_6$  met een verwerkingstijd van 4. Voor een gewone analyse krijgen we  $0.6 \cdot 4 + 0.4 \cdot (1 + 1) = 3.2$  als verwerkingstijd voor de of-splitsing. In elk geval is de tijd voor dit alternatief van de en-splitsing groter dan voor het alternatief met  $A_3$  en  $A_7$ , namelijk  $3 + 4 = 7$  in de worst-case analyse en  $3 + 3.2 = 6.2$  voor de gewone analyse.

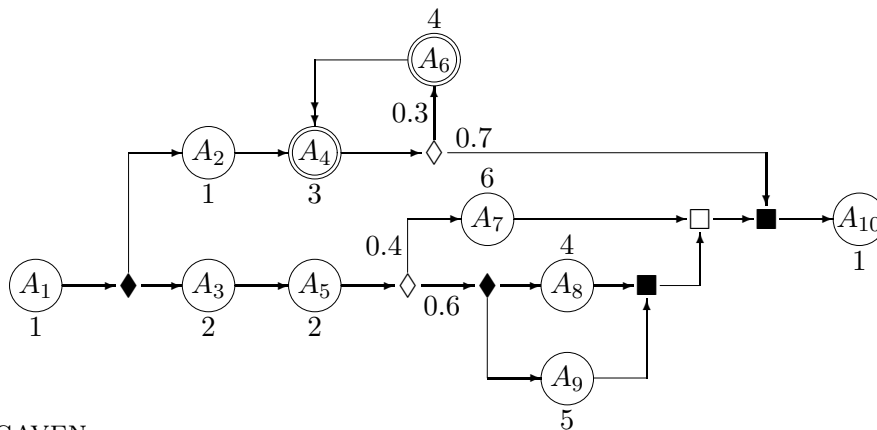
In een worst-case analyse is er dus een eenduidig kritiek pad, namelijk het pad door  $A_1$ ,  $A_2$ ,  $A_6$  en  $A_8$  en de verwerkingstijd hiervoor is  $1 + 3 + 4 + 1 = 9$ . Bij de gewone analyse moeten we ook de andere alternatieven in de of-splitsing in het kritieke pad meenemen, omdat een verandering in deze componenten ook de verwerkingstijd van het kritieke pad gaat veranderen. In dit geval zouden we dus ook  $A_4$  en  $A_5$  in het kritieke pad hebben.

Als we in dit voorbeeld één component zouden mogen verbeteren, zou  $A_6$  de meest voor de hand liggende keuze zijn. Als we dit namelijk op een verwerkingstijd van  $d_6 = 2$  kunnen reduceren, hebben we een leuk evenwicht in het

netwerk bereikt. De twee alternatieven in de of-splitsing hebben dan namelijk dezelfde verwerkingstijd en ook de verwerkingstijden voor de twee alternatieven van de en-splitsing hebben dan dezelfde tijden.

#### BELANGRIJKE BEGRIPPEN IN DEZE LES

- netwerk van acties
- en-splitsing, of-splitsing, iteratie
- convolutieproduct
- kritiek pad
- worst-case analyse, gewone analyse



#### OPGAVEN

34. Bepaal welke acties er in het kritieke pad van het bovenstaande netwerk liggen:
- bij een worst-case analyse, d.w.z. bij een of-splitsing komt alleen maar het slechtste alternatief in het kritieke pad terecht.
  - bij een gewone analyse, d.w.z. alle alternatieven van een of-splitsing komen in het kritieke pad terecht.
35. Je mag 3000€ in het netwerk investeren. Met 1000€ kan je één actie om 1 tijdseenheid versnellen. Je mag het geld over een, twee of drie acties verdelen, maar elke actie zal steeds minstens een verwerkingstijd van 1 hebben (dus aan een actie met een verwerkingstijd van 2 kan je niet meer dan 1000€ besteden).
- Wat is bij een worst-case analyse de beste keuze voor je investering, d.w.z. hoe kan je de verwerkingstijd van het kritieke pad het meeste reduceren?
  - Hoe zit het bij een gewone analyse?
36. De aangegeven verwerkingstijden zijn gemiddelde tijden. In werkelijkheid hebben de enkele acties de volgende kansverdelingen voor hun verwerkingstijden: Voor een actie met verwerkingstijd  $d$  is

$$P(t = d) = 0.5, \quad P(t = d - 1) = 0.25 \quad \text{en} \quad P(t = d + 1) = 0.25$$

(d.w.z. de aangegeven verwerkingstijden gelden voor de helft van de opdrachten, 25% zijn een tijdseenheid sneller en 25% een tijdseenheid langzamer).

- (i) Bepaal de kansverdeling voor de verwerkingstijd van de en-splitsing met acties  $A_8$  en  $A_9$ .
  - (ii) Bepaal de kansverdeling voor de verwerkingstijd van de of-splitsing met acties  $A_7$ ,  $A_8$  en  $A_9$ .
  - (iii) Bepaal de kansverdeling voor de verwerkingstijd van de iteratie met acties  $A_4$  en  $A_6$ . Hierbij mag je de iteratie na 1 iteratie afbreken.
  - (iv) Bereken dezelfde kansverdeling als in deel (iii), maar breek pas na 2 iteraties af. Vergelijk de kansverdeling met degene uit deel (iii).
37. Net als in de vorige opgave heeft een actie met aangegeven verwerkingstijd  $d$  de kansverdeling

$$P(t = d) = 0.5, \quad P(t = d - 1) = 0.25 \quad \text{en} \quad P(t = d + 1) = 0.25$$

voor de verwerkingstijden. De iteratie mag weer na 1 iteratie afgebroken worden.

- (i) Bepaal de kansverdeling voor de verwerkingstijd van het netwerk.
- (ii) In welke tijd kan je 90% van de opdrachten afhandelen?
- (iii) Bereken ook de verwachtingswaarde van de verwerkingstijd van het netwerk.