# $\mathrm{Stack}$ (oneindige recursieve specificatie)

Acties: $0$ ('push $0$'), $\underline{0}$ ('pop $0$'), $1$ ('push $1$') en $\underline{1}$ ('pop $1$').

De oneindige guarded recursieve BPA-specificatie

$$
\begin{aligned}
\mathrm{S}_\lambda &= 0 \cdot \mathrm{S}_0 + 1 \cdot \mathrm{S}_1 \\
\mathrm{S}_{\mathrm{d}\sigma} &= 0 \cdot \mathrm{S}_{0\mathrm{d}\sigma} + 1 \cdot \mathrm{S}_{1\mathrm{d}\sigma} + \underline{\mathrm{d}} \cdot \mathrm{S}_\sigma
\end{aligned}
$$

(voor alle $\mathrm{d} \in \{0, 1\}$ en voor alle strings $\sigma \in \{0, 1\}^*$)

specificeert het proces $\mathrm{Stack}$ over $\{0, 1\}$.

# Stack (eindige recursieve specificatie)

Acties: $0$, $\underline{0}$, $1$ en $\underline{1}$.

De eindige guarded recursieve BPA-specificatie

$$\begin{array}{lcl} S & = & T \cdot S \\ T & = & 0 \cdot T_0 + 1 \cdot T_1 \\ T_0 & = & \underline{0} + T \cdot T_0 \\ T_1 & = & \underline{1} + T \cdot T_1 \end{array}$$

specificeert eveneens het proces Stack over $\{0, 1\}$.

# Reguliere processen

Een procesgraaf $g$ heet **eindig** als $g$ eindig veel knopen en eindig veel pijlen heeft.

Een procesgraaf $g$ heet **regulier** (**finite-state**) als $g$ bisimilair is met een eindige procesgraaf.

**Stelling**: Zij $g_1, \ldots, g_n$ een oplossing voor een guarded recursieve specificatie over $V = \{X_1, \ldots, X_n\}$ waarvan de recursievariabelen $X_1, \ldots, X_n$ geen terminerende oplossingen kunnen hebben. Dan zijn $g_1, \ldots, g_n$ regulier.

**Gevolg**: De processen $\mathrm{Counter}$ en $\mathrm{Stack}$ kunnen niet in een enkele recursieve vergelijking worden gedefinieerd; er is een hulpvergelijking voor een terminerend proces nodig.

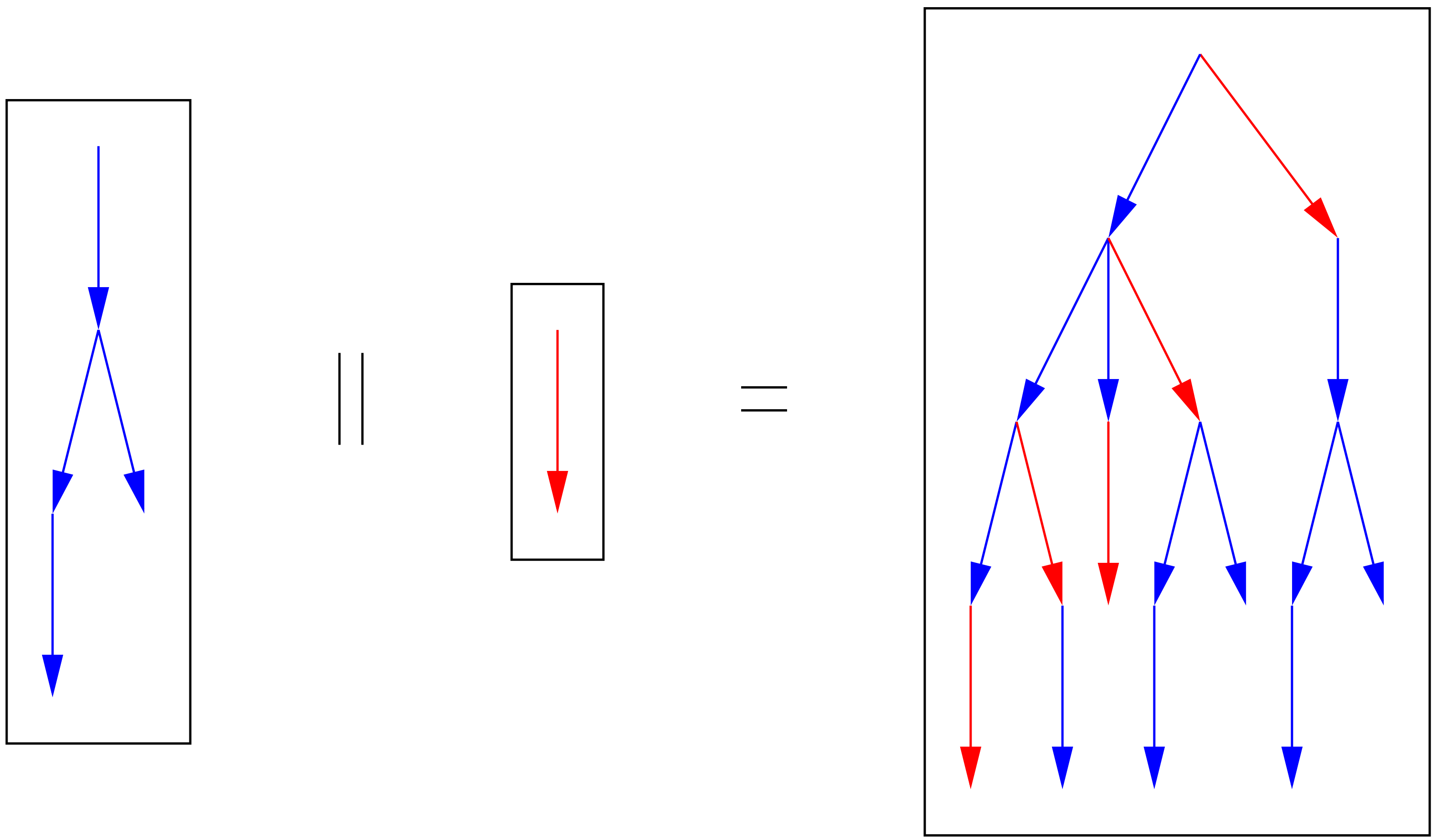
# PA: **Process Algebra (1)**

De taal van PA krijgen we door de taal van BPA uit te breiden met twee binaire operaties:

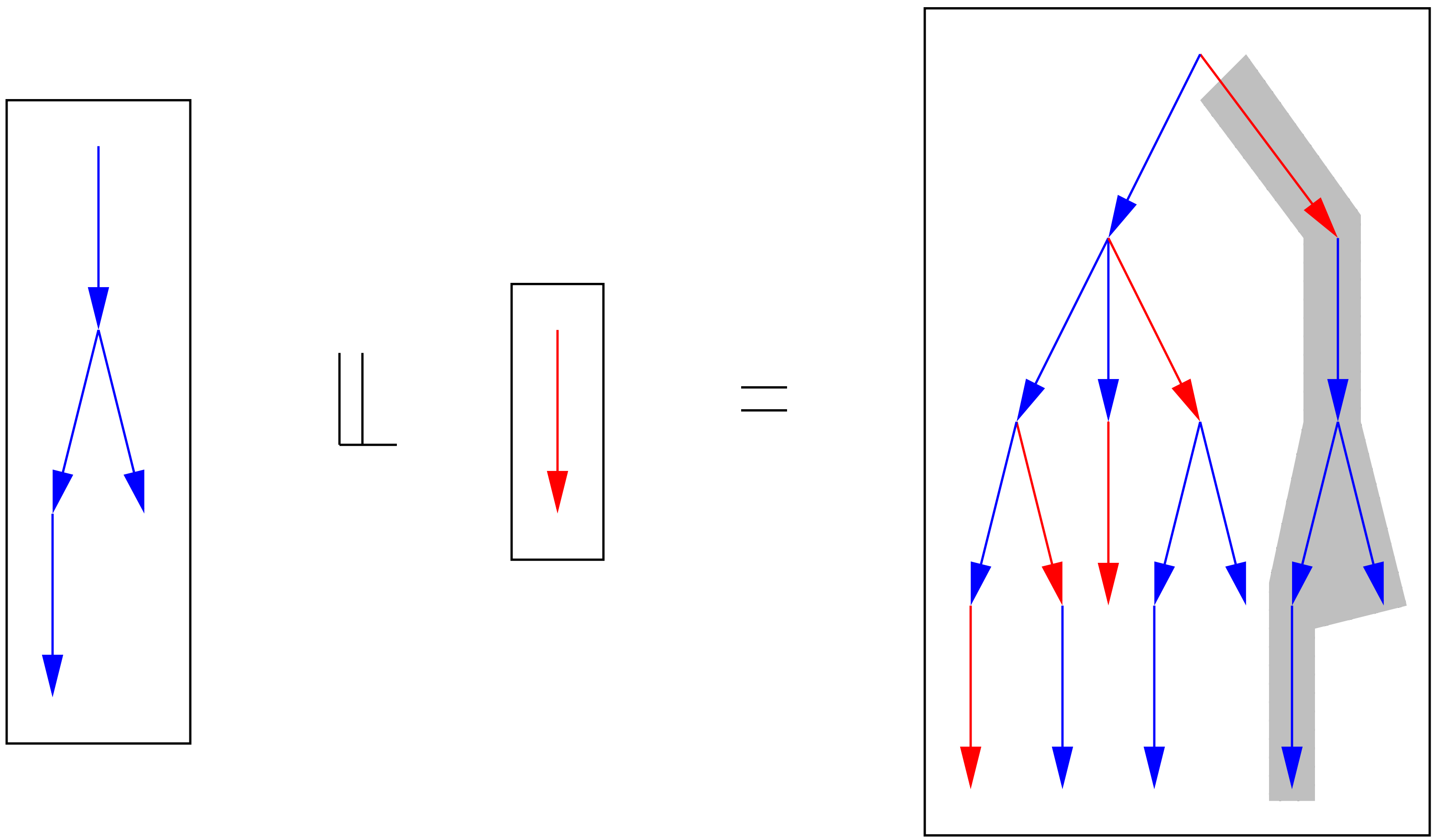
$\parallel$ : **merge** (**parallelle compositie**) van processen;
$p \parallel q$ is het proces dat de acties van $p$ en $q$ **interleave**-t;

$\mathbin{\|\!\!\_}$ : **left-merge** van processen;
$p \mathbin{\|\!\!\_} q$ is het proces dat zich gedraagt als $p \parallel q$ met de aanvullende eis dat de eerste actie van $p$ moet komen.

# Merge (informeel)



# Left-merge (informeel)



# PA: **Process Algebra (2)**

De axioma's van PA:

(A1)–(A5) (d.w.z., de axioma's van BPA) plus

$$
\begin{array}{lll}
\text{(M1)} & x \parallel y = (x \mathbin{\|\!\lfloor} y) + (y \mathbin{\|\!\lfloor} x) & \\
\text{(M2)} & a \mathbin{\|\!\lfloor} x = a \cdot x & \text{(voor alle } a \in A\text{)} \\
\text{(M3)} & (a \cdot x) \mathbin{\|\!\lfloor} y = a \cdot (x \parallel y) & \text{(voor alle } a \in A\text{)} \\
\text{(M4)} & (x + y) \mathbin{\|\!\lfloor} z = (x \mathbin{\|\!\lfloor} z) + (y \mathbin{\|\!\lfloor} x) &
\end{array}
$$

We schrijven PA $\vdash p = q$ als $p$ door 'toepassing van bovenstaande axioma's' overgaat in $q$.[1]

[1] formeel: als $p = q$ met equationele logica afleidbaar is uit de axioma's van PA.

# Eliminatie van $\parallel$ en $\mathbin{\|\!\!\lfloor}$

**Stelling**: Uit PA-termen *zonder recursievariabelen* kunnen de operaties $\parallel$ en $\mathbin{\|\!\!\lfloor}$ worden geëlimineerd, d.w.z.:

> Als $p$ een PA-term is en in $p$ komen geen recursievariabelen voor, dan is er een BPA-term $q$ (dus zonder $\parallel$ en $\mathbin{\|\!\!\lfloor}$) zo dat PA $\vdash p = q$.

**Let op**: het is in het algemeen *niet* mogelijk om de operaties $\parallel$ en $\mathbin{\|\!\!\lfloor}$ te elimineren uit recursieve PA-specificaties! Later zullen we een voorbeeld zien van een proces dat kan worden gedefinieerd met een eindige PA-specificatie, maar niet met een eindige BPA-specificatie.

# Procesgrafen (6)

Zij $\mathrm{g}_1, \mathrm{g}_2 \in \mathbf{G}(\mathrm{A})$, d.w.z., $\mathrm{g}_1$ en $\mathrm{g}_2$ zijn samenhangende procesgrafen met een acyclische wortel.

We definiëren het **cartesisch product** $\mathrm{g}_1 \parallel \mathrm{g}_2$ van $\mathrm{g}_1$ en $\mathrm{g}_2$ als de procesgraaf met:

(i) als knopen alle paren $(s_1, s_2)$ met $s_1$ een knoop van $\mathrm{g}_1$ en $s_2$ een knoop van $\mathrm{g}_2$;

(ii) als pijlen alle $(s_1, s_2) \longrightarrow_{\mathrm{a}} (t_1, t_2)$ waarvoor geldt dat:

- ofwel $s_1 \longrightarrow_{\mathrm{a}} t_1$ en $s_2 = t_2$,
- ofwel $s_1 = t_1$ en $s_2 \longrightarrow_{\mathrm{a}} t_2$; en

(iii) als wortel het paar $(r_1, r_2)$ waarin $r_1$ de wortel van $\mathrm{g}_1$ en $r_2$ de wortel van $\mathrm{g}_2$.

# Procesgrafen (7)

Zij $\mathrm{g}_1, \mathrm{g}_2 \in \mathbf{G}(\mathrm{A})$, d.w.z., $\mathrm{g}_1$ en $\mathrm{g}_2$ zijn samenhangende procesgrafen met een acyclische wortel.

We definiëren $\mathrm{g}_1 \mathbin{|\!\lfloor} \mathrm{g}_2$ als de procesgraaf die uit $\mathrm{g}_1 \parallel \mathrm{g}_2$ wordt verkregen door:

(i) alle pijlen van de vorm $(r_1, r_2) \rightarrow_{\mathrm{a}} (r_1, s)$, met $(r_1, r_2)$ de wortel van $\mathrm{g}_1 \mathbin{|\!\lfloor} \mathrm{g}_2$ en $s$ een willekeurige knoop van $\mathrm{g}_2$, te verwijderen;

(ii) alle daardoor onbereikbaar geworden delen te verwijderen.

# Correctheid en volledigheid

We krijgen de interpretatie van PA-termen in $\mathbf{G}(\mathrm{A})$ door de interpretatie van BPA-termen in $\mathbf{G}(\mathrm{A})$ uit te breiden met

$$\mathbf{G}(p \parallel q) = \mathbf{G}(p) \parallel \mathbf{G}(q)$$
$$\mathbf{G}(p \mathbin{\|\!\_} q) = \mathbf{G}(p) \mathbin{\|\!\_} \mathbf{G}(q).$$

**Stelling**: Voor alle gesloten PA-termen $p$ en $q$:

$$\mathsf{PA} \vdash p = q \iff \mathbf{G}(p) \,\underline{\leftrightarrow}\, \mathbf{G}(q).$$