

# Utilisation d'opérateurs pour le calcul dans les réseaux

Bertrand Ducourthial

UMR CNRS 6599 HEUDIASYC

Université de Technologie de Compiègne

`Bertrand.Ducourthial@hds.utc.fr`



# Calcul dans un réseau

- Réseau d'«entités calculantes»
  - calculent et communiquent
  - concurrent à la réalisation d'une tâche globale



# Calcul dans un réseau

- Réseau d'«entités calculantes»
- Tâches dynamiques
  - résultat = exécution (infinie)
  - pas de terminaison
  - exemple : exclusion mutuelle



# Calcul dans un réseau

- Réseau d'«entités calculantes»
- Tâches dynamiques
- Tâches statiques
  - résultat = configuration (état du réseau et des entités)
  - silencieux
  - exemple : calcul de distance



# Calcul dans un réseau

- Réseau d'«entités calculantes»
- Tâches dynamiques
- Tâches statiques
- Formulation du calcul**
  - algorithme local pour chaque entité
  - événement/condition  $\Rightarrow$  instructions
  - garde  $\longrightarrow$  liste d'instructions
  - programme = liste d'actions gardées



# Calcul dans un réseau

- Réseau d'«entités calculantes»
- Tâches dynamiques
- Tâches statiques
- Formulation du calcul
- Calcul dans un réseau
  - initialiser le calcul
  - lancer le calcul
  - récupérer le résultat (état sur chaque entité)
    - réseau synchrone : attente d'un délai
    - réseau asynchrone : attente de la terminaison
      - ↪ le calcul doit terminer
      - ↪ il faut détecter la terminaison



# Principe général

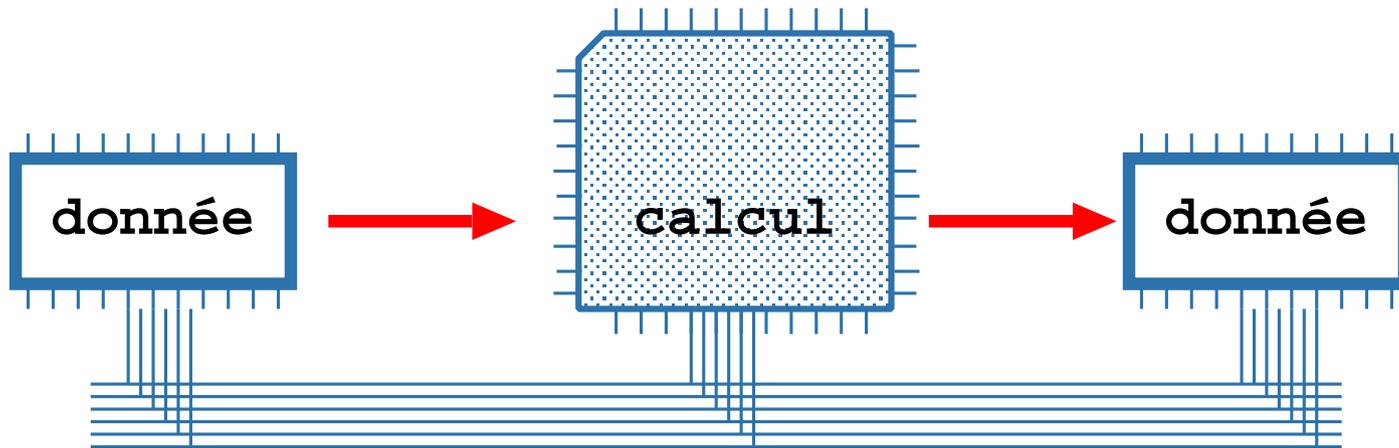
## ■ Transformation de données

donnée  calcul  donnée



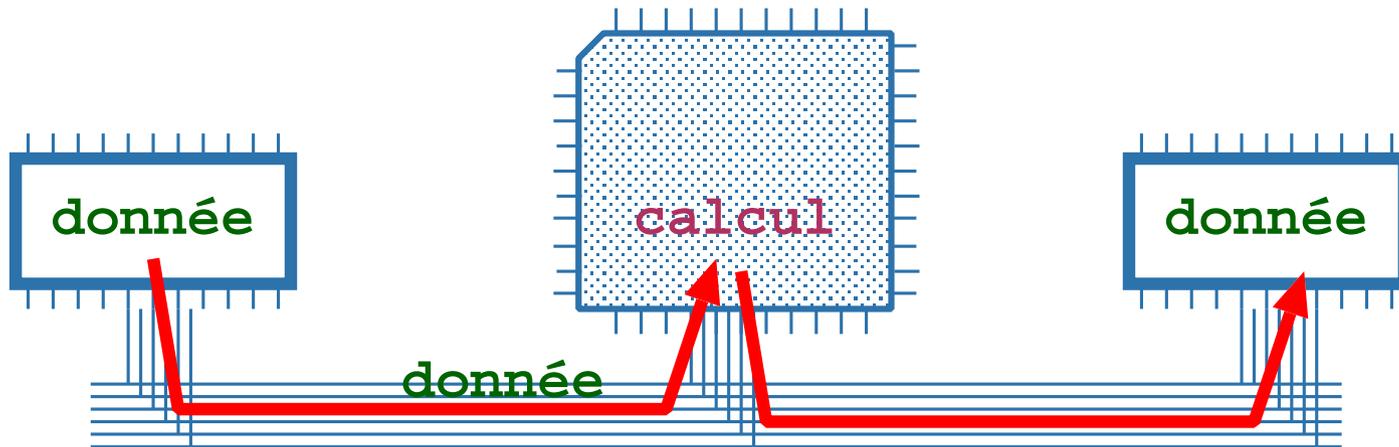
# Principe général

## ■ Transformation de données



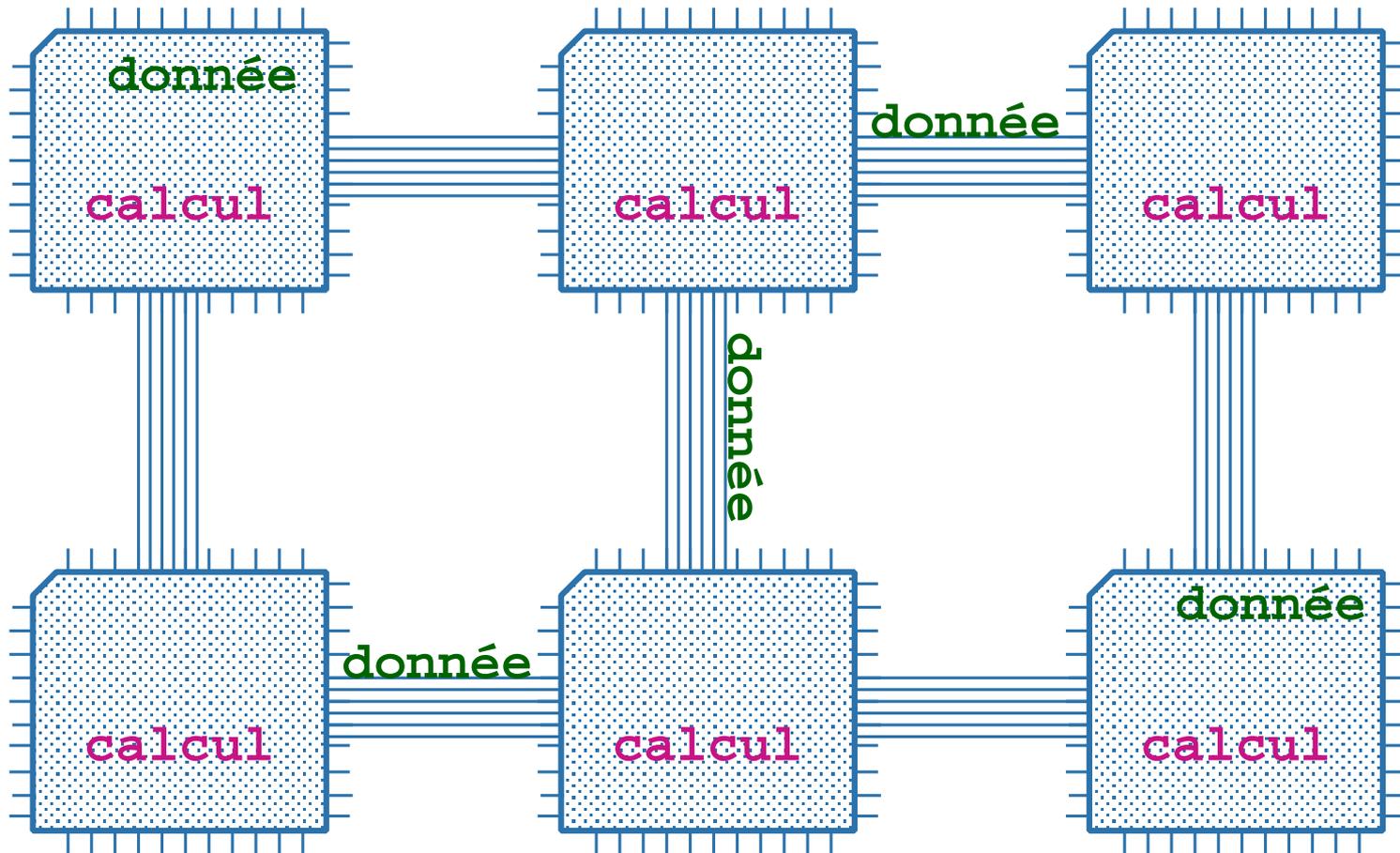
# Principe général

## ■ Transformation de données



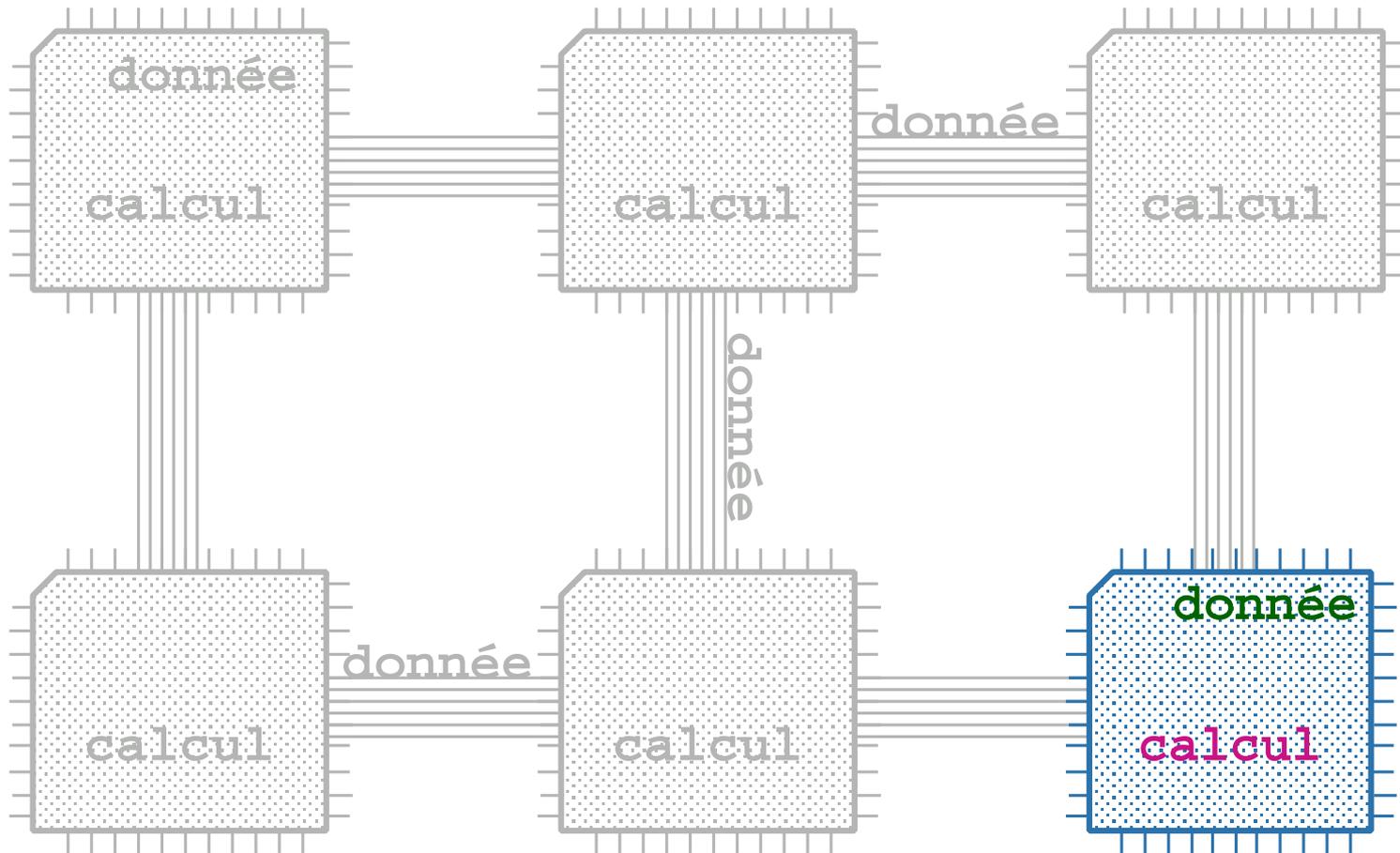
# Principe général

## ■ Transformation de données



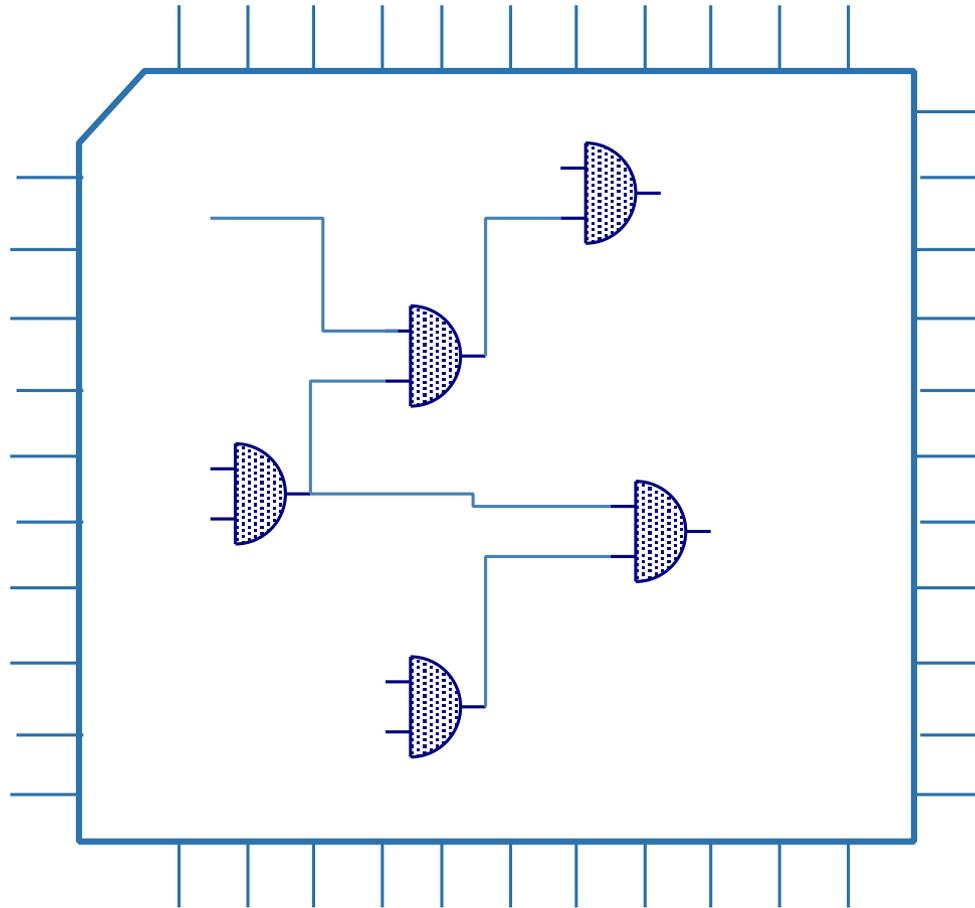
# Principe général

## ■ Transformation de données



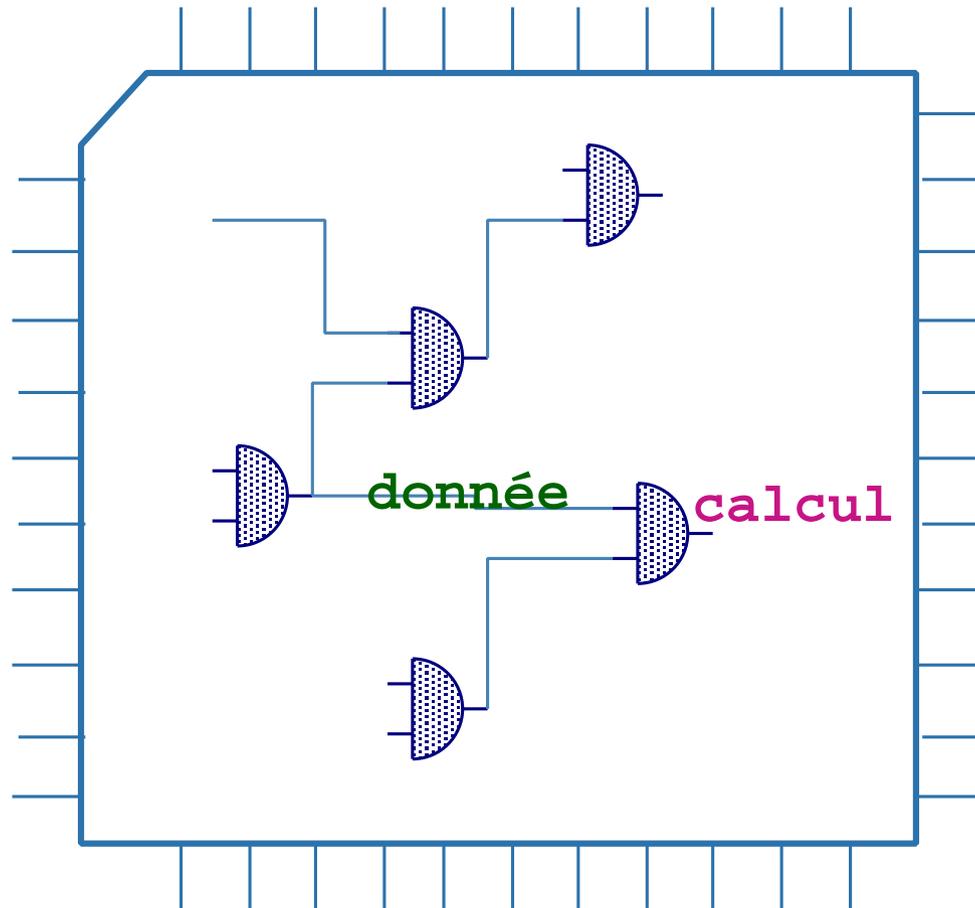
# Principe général

## ■ Transformation de données



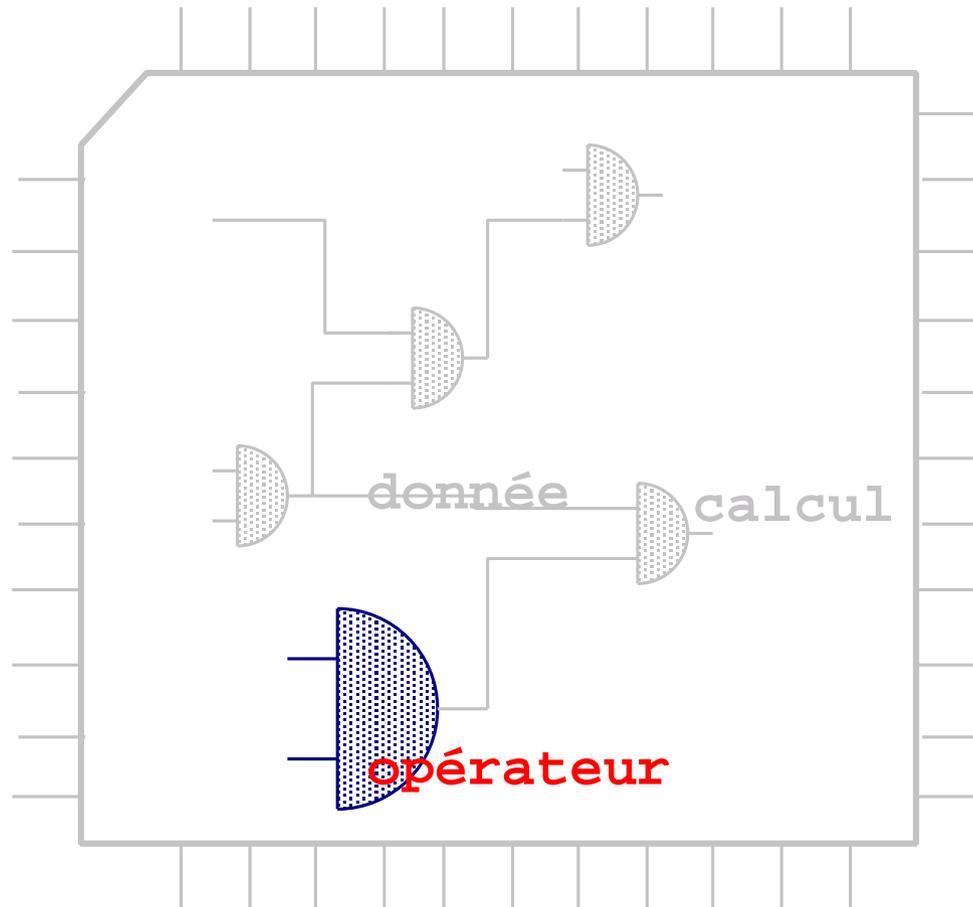
# Principe général

## ■ Transformation de données



# Principe général

## ■ Transformation de données



# Calcul $\equiv$ opérateur

## ■ Calcul préfixe

- $\mathbb{S}^n \longrightarrow \mathbb{S}$   
 $(x_1, \dots, x_n) \mapsto x_1 \diamond \dots \diamond x_n$



# Calcul $\equiv$ opérateur

## ■ Calcul préfixe

- $\mathbb{S}^n \longrightarrow \mathbb{S}$   
 $(x_1, \dots, x_n) \mapsto x_1 \diamond \dots \diamond x_n$
- $\mathbb{S}^n \longrightarrow \mathbb{S}^n$   
 $(x_1, \dots, x_n) \mapsto (x_1, x_1 \diamond x_2, \dots, x_1 \diamond \dots \diamond x_n)$

Lakshminarayanan94, Hillis85, Blueloch90, Cohn90,  
Chung95...

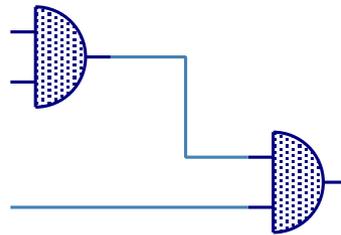


# Calcul $\equiv$ opérateur

□ Calcul préfixe

■ Circuit préfixe

- $n = 3$  entrées, taille (coût) = 2, profondeur (temps) = 2



- constructions en profondeur =  $\lfloor \log_2(n) \rfloor$  et taille  $< 4n$   
[LadnerFischer80]
- profondeur + taille  $\geq 2n - 2$   
[Snir86]

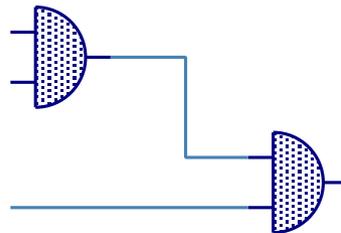
# Calcul $\equiv$ opérateur

Calcul préfixe

Circuit préfixe

Structure algébrique sous-jacente

- assemblage quelconque des sous-circuits  
 $\rightsquigarrow$   $\diamond$  associative et commutative  $\rightsquigarrow$  monoïde abélien
- relations circuit  $\leftrightarrow$  monoïde sous-jacent
- exemple : éléments donnant une constante avec une addition par la gauche  $\rightsquigarrow$  accélération



- accélération pour l'addition de mots de  $n$  bits  
pas d'accélération pour la parité binaire



# Calcul $\equiv$ opérateur

- Calcul préfixe
- Circuit préfixe
- Structure algébrique sous-jacente
- Circuits électroniques asynchrones
  - convergence du calcul
  - terminaison à détecter



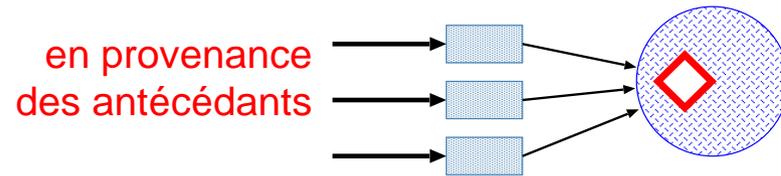
# Calcul $\equiv$ opérateur

- Calcul préfixe
- Circuit préfixe
- Structure algébrique sous-jacente
- Circuits électroniques asynchrones
- et pour un réseau d'ordinateurs ?



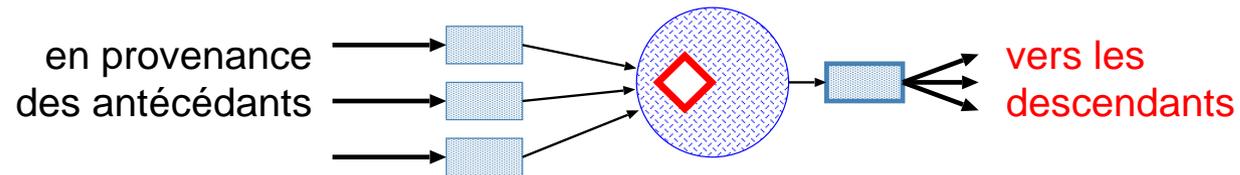
# Calcul dans un réseau

- Algorithme local  $\equiv$  opérateur



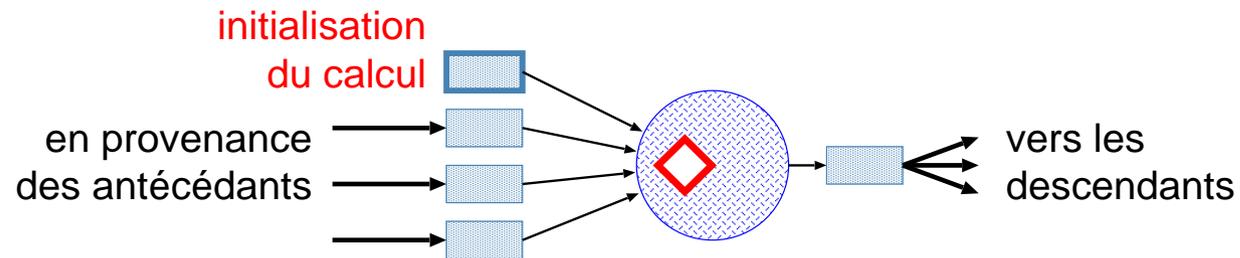
# Calcul dans un réseau

- Algorithme local  $\equiv$  opérateur



# Calcul dans un réseau

- Algorithme local  $\equiv$  opérateur



# Calcul dans un réseau

- Algorithme local  $\equiv$  opérateur
- Propriétés de l'opérateur
  - pas d'influence du câblage sur le résultat
    - ↪  $\diamond$  associatif et commutatif



# Calcul dans un réseau

□ Algorithme local  $\equiv$  opérateur

■ Propriétés de l'opérateur

- pas d'influence du câblage sur le résultat  
     $\rightsquigarrow$   $\diamond$  associatif et commutatif
- tâche statique  $\rightsquigarrow$  convergence vers un résultat par entité



# Calcul dans un réseau

□ Algorithme local  $\equiv$  opérateur

■ Propriétés de l'opérateur

- pas d'influence du câblage sur le résultat  
     $\rightsquigarrow$   $\diamond$  associatif et commutatif
- tâche statique  $\rightsquigarrow$  convergence vers un résultat par entité
- globalement, relation d'ordre sur les configurations : diminution en s'approchant de la configuration terminale



# Calcul dans un réseau

□ Algorithme local  $\equiv$  opérateur

■ Propriétés de l'opérateur

- pas d'influence du câblage sur le résultat  
     $\rightsquigarrow$   $\diamond$  associatif et commutatif
- tâche statique  $\rightsquigarrow$  convergence vers un résultat par entité
- globalement, relation d'ordre sur les configurations : diminution en s'approchant de la configuration terminale
- localement, nécessité de l'idempotence à partir d'un certain ordre



# Calcul dans un réseau

- Algorithme local  $\equiv$  opérateur
- Propriétés de l'opérateur
- Infimum**
  - associatif, commutatif, idempotent, élt. neutre
  - exemple : min, max...
  - définissent une relation d'ordre locale



# Calcul dans un réseau

- Algorithme local  $\equiv$  opérateur
- Propriétés de l'opérateur
- Infimum
- Calcul d'infimum
  - initialisation du résultat : une valeur par entité (ROM)
  - itérer le calcul local de l'infimum (e.g., minimum)
  - détecter la terminaison
  - convergence vers le minimum par composante fortement connexe
  - *cf. waves algorithms* avec plusieurs initiateurs [Tel91]



# Calcul dans un réseau

- Algorithme local  $\equiv$  opérateur
  - Propriétés de l'opérateur
  - Infimum
  - Calcul d'infimum
  - Autres opérateurs ?
    - les infimum conviennent : min, max,  $\cup$ ,  $\cap$ , ppcm, pgcd,  $\wedge$ ,  $\vee$ , ...
    - un résultat identique par composante fortement connexe
- $\rightsquigarrow$  autres opérateurs ?



# $r$ -opérateurs

## ■ infimum ( $s$ -opérateur)

associatif	$(x \oplus y) \oplus z = x \oplus (y \oplus z)$
commutatif	$x \oplus y = y \oplus x$
idempotent	$x \oplus x = x$
élément neutre	$x \oplus e_{\oplus} = x$



# $r$ -opérateurs

□ infimum ( $s$ -opérateur)

■  $r$ -opérateur

$\triangleleft$  est un  $r$ -opérateur sur  $\mathbb{S}$  s'il existe une bijection  $r : \mathbb{S} \rightarrow \mathbb{S}$  telle que  $\triangleleft$  est :

$r$ -associatif	$(x \triangleleft y) \triangleleft r(z) = x \triangleleft (y \triangleleft z)$
$r$ -commutatif	$r(x) \triangleleft y = r(y) \triangleleft x$
$r$ -idempotent	$r(x) \triangleleft x = r(x)$
neutre à droite	$x \triangleleft e_{\triangleleft} = x$



# $r$ -opérateurs

- infimum ( $s$ -opérateur)
- $r$ -opérateur
- Construction d'un  $r$ -opérateur
  - $s$ -opérateur  $\oplus$  tel que  $\min, \max, \cup, \cap, \wedge, \vee, \dots$
  - application  $r$  sur l'ensemble de déf. de  $\oplus$  (bijective)



# $r$ -opérateurs

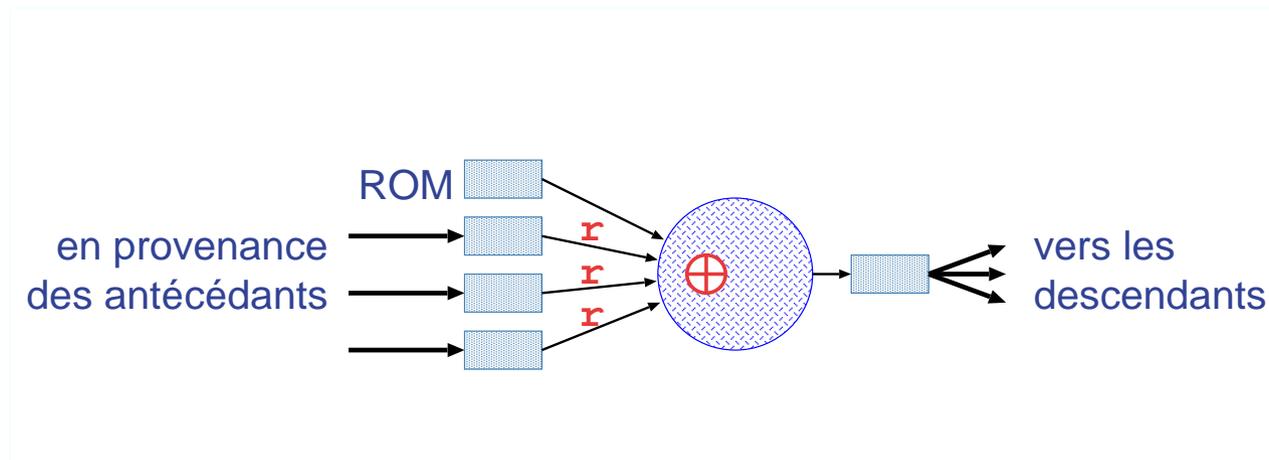
infimum ( $s$ -opérateur)

$r$ -opérateur

Construction d'un  $r$ -opérateur

- $s$ -opérateur  $\oplus$  tel que  $\min, \max, \cup, \cap, \wedge, \vee, \dots$

- application  $r$  sur l'ensemble de déf. de  $\oplus$  (bijective)



# $r$ -opérateurs

- infimum ( $s$ -opérateur)
- $r$ -opérateur
- Construction d'un  $r$ -opérateur

## ■ Exemples

- $\text{minc}(x, y) = \min(x, y + 1)$
- $\text{maxmul}(x, y) = \max(x, y)$
- $\text{lexicat}_e((a, d, f), (b, d)) = (a, d, f) \oplus (b, d, e) = (a, b, e)$
- $\text{maxmin}_\kappa(x, y) = \max(x, \min(y, \kappa))$
- ...

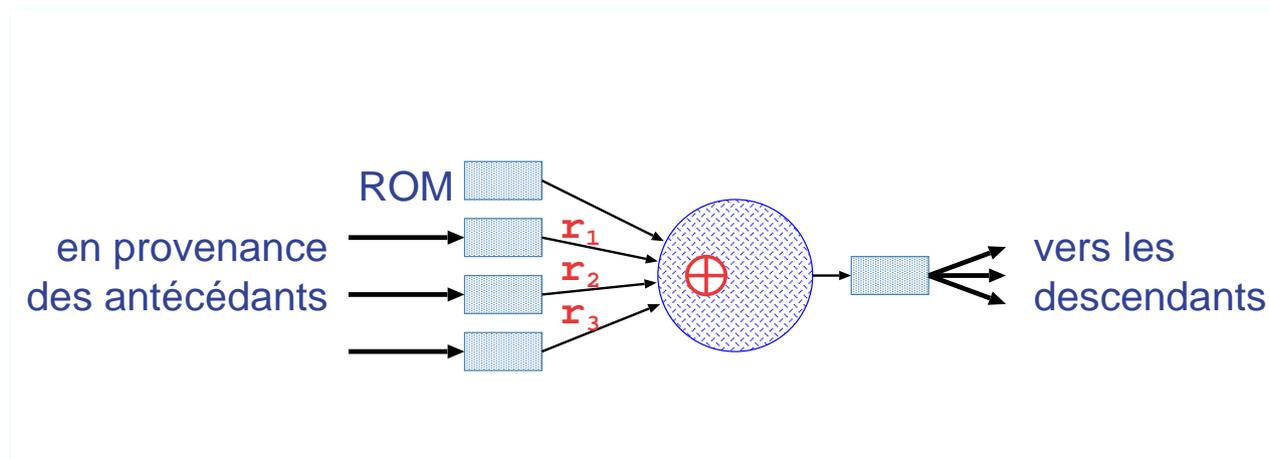


# $r$ -opérateurs

- infimum ( $s$ -opérateur)
- $r$ -opérateur
- Construction d'un  $r$ -opérateur
- Exemples

## ■ $r$ -opérateurs $n$ -aires

- $\text{minc}_w(x_0, \dots, x_n) = \min(x_0, x_1 + w_1, \dots, x_n + w_n)$
- $\text{maxmul}_\pi(x_0, \dots, x_n) = \max(x_0, x_1 \times \pi_1, \dots, x_n \times \pi_n)$



# $r$ -opérateurs

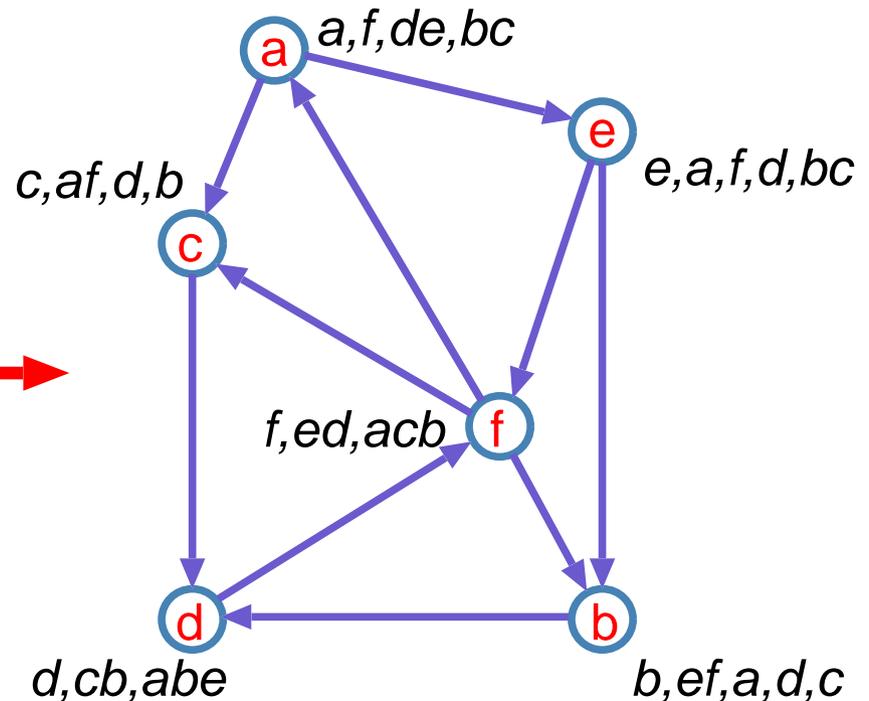
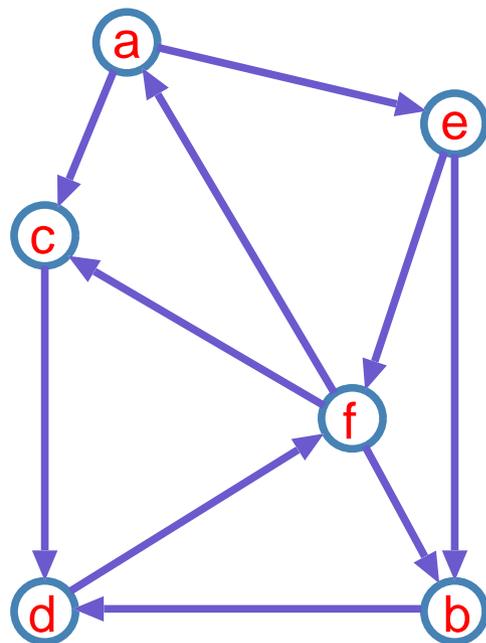
- infimum ( $s$ -opérateur)
- $r$ -opérateur
- Construction d'un  $r$ -opérateur
- Exemples
- $r$ -opérateurs  $n$ -aires
- Applications
  - composante fort. connexe  $\rightsquigarrow$  distance  $\rightsquigarrow$  chemin
  - forêt des chemins les plus courts, les plus fiables, de meilleure capacité *etc.*
  - arbre en profondeur
  - liste des antécédants
  - ...



# Exemple d'utilisation

## Liste des antécédants, ordonnée par la distance

- tâche statique
- initialisation : l'identité du nœud  $(\{v\})$
- traitement locaux
- résultat final : liste ordonnée des antécédants



# Exemple d'utilisation

- Liste des antécédants, ordonnée par la distance
- ensemble  $\mathcal{S}$  des données manipulées
  - ensemble des listes d'ensemble de sommets  
exemple :  $(\{d\}, \{b\}, \{a, c\})$



# Exemple d'utilisation

□ Liste des antécédants, ordonnée par la distance

■ ensemble  $\mathcal{S}$  des données manipulées

- ensemble des listes d'ensemble de sommets  
exemple :  $(\{d\}, \{b\}, \{a, c\})$

- éliminer les éléments indésirables de  $\mathcal{S}_1$  :

$$(\{a, b\}, \{c\}, \{d, a\}, \{c\}, \emptyset, \{e\}) \equiv$$

$$(\{a, b\}, \{c\}, \{d, \cancel{a}\}, \cancel{\{c\}}, \emptyset, \cancel{\{e\}}) \equiv (\{a, b\}, \{c\}, \{d\})$$

$$\mathcal{S} = \mathcal{S}_1 / \equiv$$



# Exemple d'utilisation

Liste des antécédants, ordonnée par la distance

ensemble  $\mathcal{S}$  des données manipulées

$s$ -opérateur

- calcul : fabrication d'une liste avec les listes reçues

- fusion des contributions des différents antécédants :

$$(\{d\}, \{b\}, \{a, c\}) \oplus (\{c\}, \{a, e\}, \{b\}) = (\{d, c\}, \{b, a, e\}, \{a, c, b\})$$



# Exemple d'utilisation

□ Liste des antécédants, ordonnée par la distance

□ ensemble  $\mathcal{S}$  des données manipulées

■  $s$ -opérateur

- calcul : fabrication d'une liste avec les listes reçues

- fusion des contributions des différents antécédants :

$$(\{d\}, \{b\}, \{a, c\}) \oplus (\{c\}, \{a, e\}, \{b\}) = (\{d, c\}, \{b, a, e\}, \{a, c, b\})$$

- épuration de la fusion :

$$(\{d, c\}, \{b, a, e\}, \{\cancel{a}, \cancel{c}, \cancel{b}\}) = (\{d, c\}, \{b, a, e\})$$



# Exemple d'utilisation

□ Liste des antécédants, ordonnée par la distance

□ ensemble  $\mathcal{S}$  des données manipulées

■  $s$ -opérateur

- calcul : fabrication d'une liste avec les listes reçues

- fusion des contributions des différents antécédants :

$$(\{d\}, \{b\}, \{a, c\}) \oplus (\{c\}, \{a, e\}, \{b\}) = (\{d, c\}, \{b, a, e\}, \{a, c, b\})$$

- épuration de la fusion :

$$(\{d, c\}, \{b, a, e\}, \{a, c, b\}) = (\{d, c\}, \{b, a, e\})$$

→  $\oplus$  fusion terme à terme avec suppression des doubles



# Exemple d'utilisation

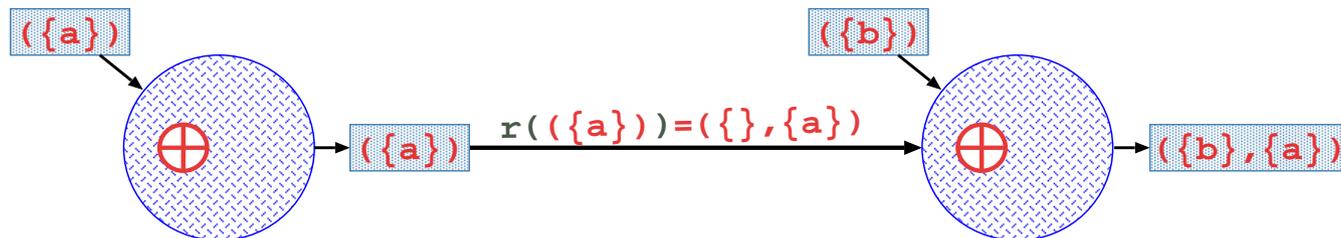
- Liste des antécédants, ordonnée par la distance
- ensemble  $\mathcal{S}$  des données manipulées
- $s$ -opérateur
- $r$ -fonction
  - sans  $r$ -fonction, un seul élément par liste
  - nécessité d'un décalage dans les listes
  - $r ( (\{d\}, \{b\}, \{a, c\}) ) = ( \{\emptyset, \{d\}, \{b\}, \{a, c\} \} )$



# Exemple d'utilisation

- Liste des antécédants, ordonnée par la distance
- ensemble  $\mathcal{S}$  des données manipulées
- $s$ -opérateur
- $r$ -fonction
- $r$ -opérateur

$$\begin{aligned} \triangleleft : \mathcal{S}^2 &\rightarrow \mathcal{S} \\ (x, y) &\mapsto x \oplus r(y) \end{aligned}$$



# Intérêt des opérateurs

## ■ Généricité

- formulation générique
- réutilisation de propriétés démontrées pour une classe d'opérateurs



# Intérêt des opérateurs

Généricité

Localité

- vérification de conditions locales aux nœuds
- propriétés globales au système



# Intérêt des opérateurs

Généricité

Localité

Propriétés de convergence

●  $r$ -opérateur idempotent  $\rightsquigarrow$  convergence

●  $\min(x, y)$ ,  $\text{minc}(x, y) = \min(x, y+1)$

$\rightsquigarrow$  oui

●  $\min(x, y-1)$

$\rightsquigarrow$  non



# Intérêt des opérateurs

- Généricité
- Localité
- Propriétés de convergence
- Propriétés de tolérance aux défaillances
  - $r$ -opérateur strictement idempotent  $\rightsquigarrow$  convergence
  - hypothèses de synchronisation  $\leftrightarrow \oplus$



# Intérêt des opérateurs

- Généricité
- Localité
- Propriétés de convergence
- Propriétés de tolérance aux défaillances
- Modélisation matricielle**
  - convergence d'une itération matricielle asynchrone
  - revient à vérifier la convergence d'une it. synchrone
  - permet d'étendre les propriétés auto-stabilisantes
  - vers une étude de la vitesse de convergence



# Conclusion

## ■ Résumé

- Calcul d'opérateurs dans un réseau
- Application dans les systèmes répartis
  - algorithme local  $\equiv$  opérateur
  - propriétés de l'opérateur, infimum
  - $r$ -opérateur
- Exemple d'utilisation
- Intérêt
  - généricité
  - seulement des propriétés locales à vérifier
  - propriétés de terminaison et d'auto-stabilisation

## ■ Perspectives

- formuler d'autres tâches statiques
- étudier d'autres propriétés génériques

