

Détection de la terminaison
 MA - SDRP
 LaBRI
 ENSEIRB - Université Bordeaux 1
<http://www.labri.fr/visidia/>

VISIDIA 1

Calcul d'un arbre recouvrant séquentiel avec détection de la terminaison

R1 > R2

VISIDIA 2

Distribué avec terminaison

VISIDIA 3

Problème de la terminaison

- Quand un process décide que le calcul a globalement terminé ?
- Problème difficile et important en calcul distribué
- Problèmes liés
 - état global consistant
 - deadlocks
 - ramasse-miettes distribués

VISIDIA 4

Définitions

- **Configuration terminale d'un algorithme:** le calcul distribué a fini (aucune modification du réseau)
- **Etat terminal d'un process:** aucun événement ne se produit sur le process
- Il est possible d'avoir une configuration terminale sans que certains process ne soient dans état terminal ([terminaison implicite](#))
- **terminaison explicite:** configuration terminale + état terminal de chaque process.

VISiDiA

5

Algorithmes de détection de la terminaison

Déf: tous les processeurs ont terminé et aucun message n'est en transit.

Terminaison explicite: Il existe au moins un processeur qui détecte la terminaison globale.

/ Algorithmes de Dijkstra-Scholten (diffusion)

- Maintien d'un arbre dynamique de processeurs actifs.
- Chaque processeur a deux états: passif, actif, et un compteur (du nombre de fils activés).
- Initialement : un processeur actif (racine), tous les autres sont passifs.

VISiDiA

6

Suite Dijkstra-Scholten

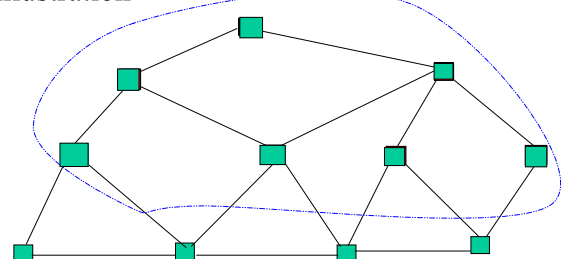
- **Principe:**

- Chaque processeur actif peut « activer » ses voisins passifs (qui deviennent ses fils s'ils sont activés pour la première fois)
- Le passage d'un état actif a passif est spontané.
- Un processeur passif et qui n'a aucun descendant actif (feuille de l'arbre) envoie un acquittement à son père dans l'arbre.
- Terminaison: racine est passive et ne possède aucun descendant.

VISiDiA

7

Illustration



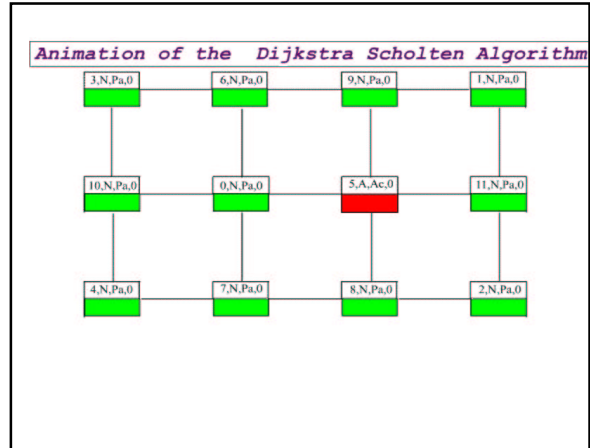
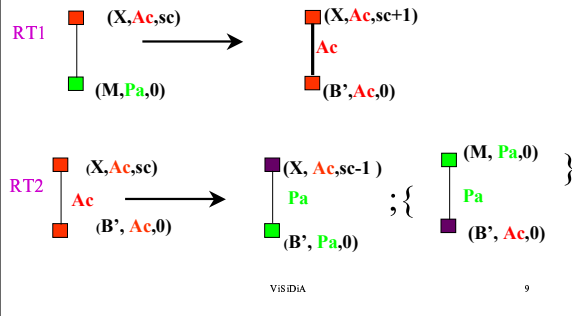
Terminé !!!!

VISiDiA

8

Un système de réécriture décrivant Dijkstra-Scholten

- Chaque sommet : une étiquette (X,Y,sc)
 X : marquage $\{A,A'\}$; Y : actif ou passif $\{Ac,Pa\}$



Applications:

Preuve de l'algorithme: Les invariants:

- $sc+1$ est le nombre d'arêtes étiquetées Ac incidentes au sommet étiqueté (X,Y,sc) .
- Toutes les arêtes incidentes à $(N,Pa,0)$ sont étiquetées Pa .
- Le sous graphe induit par les sommets étiquetés (X,Y,sc) et les arêtes étiquetées Ac est connexe, et ne possède pas de cycle.

La démonstration de ces invariants se fait par récurrence sur les règles de réécriture.

II/ Algorithmes de Dijkstra-Feijen-Van Gasteren (sondes).

- Graphe ayant un circuit hamiltonien (avec un sommet initial).
- Jeton qui circule pour signaler l'activité des sommets (sonde)
- La terminaison est détectée s'il fait un tour complet et si tous les sommets sont passifs
- Il est possible de décrire cet algorithme par un système de séécriture.
- Application à la terminaison.

III/Terminaison par SSP (par des prédicats) (Szymanski, Shi et Prywes)

- A chaque sommet $v \rightarrow$ un prédicat $p(v)$ et un entier $a(v)$

- Initialement : $p(v) := \text{faux}$

$$a(v) := -1$$

Calcul : si v a pour voisins v_1, v_2, \dots, v_d

Si $p(v) = \text{faux}$ alors $a(v) = -1$;

$p(v) = \text{vrai}$ alors $a(v) = 1 + \text{Min}\{a(v_k) \mid k \text{ varie entre } 1 \text{ et } d\}$

SSP permet de savoir que p est vraie pour une boule de centre v et de rayon $a(v)$.

Application: terminaison.

VISiDiA

13

Ajout de la terminaison à un algorithme distribué

- Si A est un algorithme distribué codé par R_A
(algorithme principal)
- Si T est un algorithme de détection de terminaison codé par R_T
(algorithme de terminaison)
- La combinaison de R_A et R_T code A avec la propriété de détection de la terminaison.

VISiDiA

14

Formulation par des systèmes de réécriture

- un produit particulier :

- nouvel étiquetage (étiquettes de R_A , étiquettes de R_T)

- union des deux ensembles de règles +

- règles de terminaison (avec contexte interdit):

“application des règles de terminaison si aucune règles de l’algorithme principal n’est applicable”

VISiDiA

15

EXEMPLES:

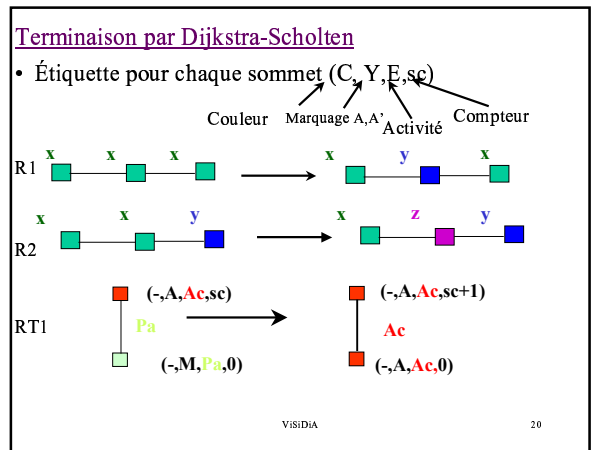
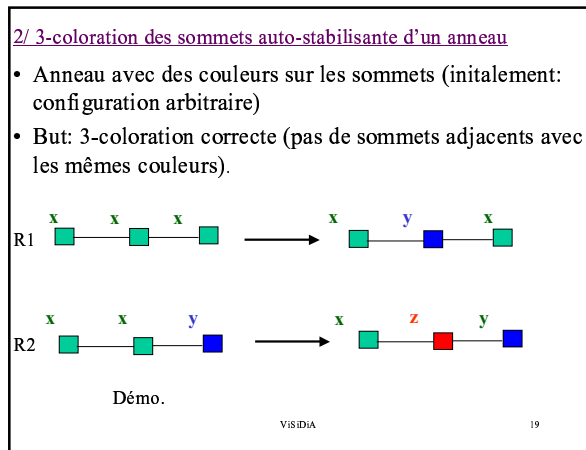
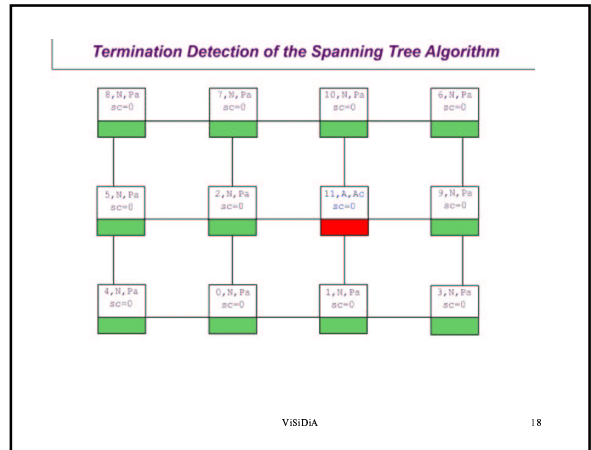
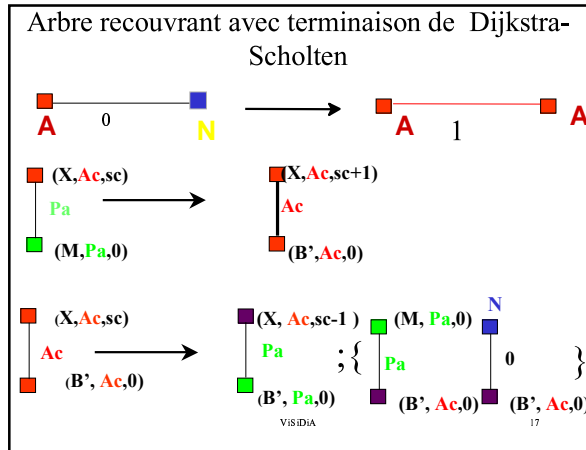
1/ Application à la détection de la terminaison du calcul d'un arbre recouvrant

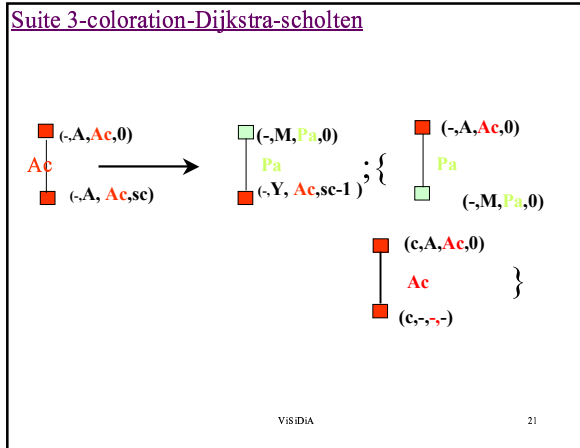
- Un système de réécriture décrivant un algorithme distribué ou le calcul est initialisé par un seul sommet.
- Pour le rendre détecter localement la terminaison, il suffit de « superposer » deux systèmes de réécritures.
- Exemple: le calcul d'un arbre recouvrant



VISiDiA

16





Exercice

- Donner un système de réécriture pour détecter la terminaison de la 3-coloration avec SSP

VISIDIA 22

