Partie 2 – Problèmes de satisfaction de contraintes

2.2 LA RESOLUTION

2.2.3 LES METHODES PROSPECTIVES

Les méthodes prospectives

- Les méthodes prospectives mettent en œuvre une étape de propagation de contraintes à chaque nœud de l'arbre de recherche
  - On supprime des valeurs sans support à des variables
  - Si un domaine devient vide alors « bakctracking » immédiat et restauration des domaines

- MAC (maintenir l'arc-consistance) [Sabin et Freuder, 1994]
  propage en lançant un algo de restauration de l'arc-consistance après chaque assignation (et aussi après chaque échec d'assignation)
  - Si x est la variable qui vient d'être assignée, il suffit d'initialiser le Q de AC3 à $\{(x',c) \mid c \in R.C, x \in P(c), x' \in P(c), x' \neq x\}$

- FC (forward checking) binaire [Haralick & Eliott, 1980] et n-aire [Van Hentenryck,1989]
  ne supprime à chaque assignation que les valeurs sans tuple support des variables adjacentes à la variable assignée

MAC

<u>Fonction</u> MAC(Network R=(X,D,C)) : Booléen

<u>Début</u>

Rendre R Arc-Consistance;

<u>si</u> R contient un domaine vide <u>alors</u> <u>retourner</u> false;

<u>si</u> R est complètement instancié <u>alors</u>

<u>afficher D ;</u>

<u>retourner</u> true;

<u>finsi;</u>

x ← ChoixVariableNonAssignée(X);

v ← ChoixValeur(D(X));

<u>si</u> MAC((X,D\D(x)∪{(x,v)},C)) <u>alors</u> <u>retourner</u> true;

MAC((X,D\{(x,v)},C));

<u>Fin</u>

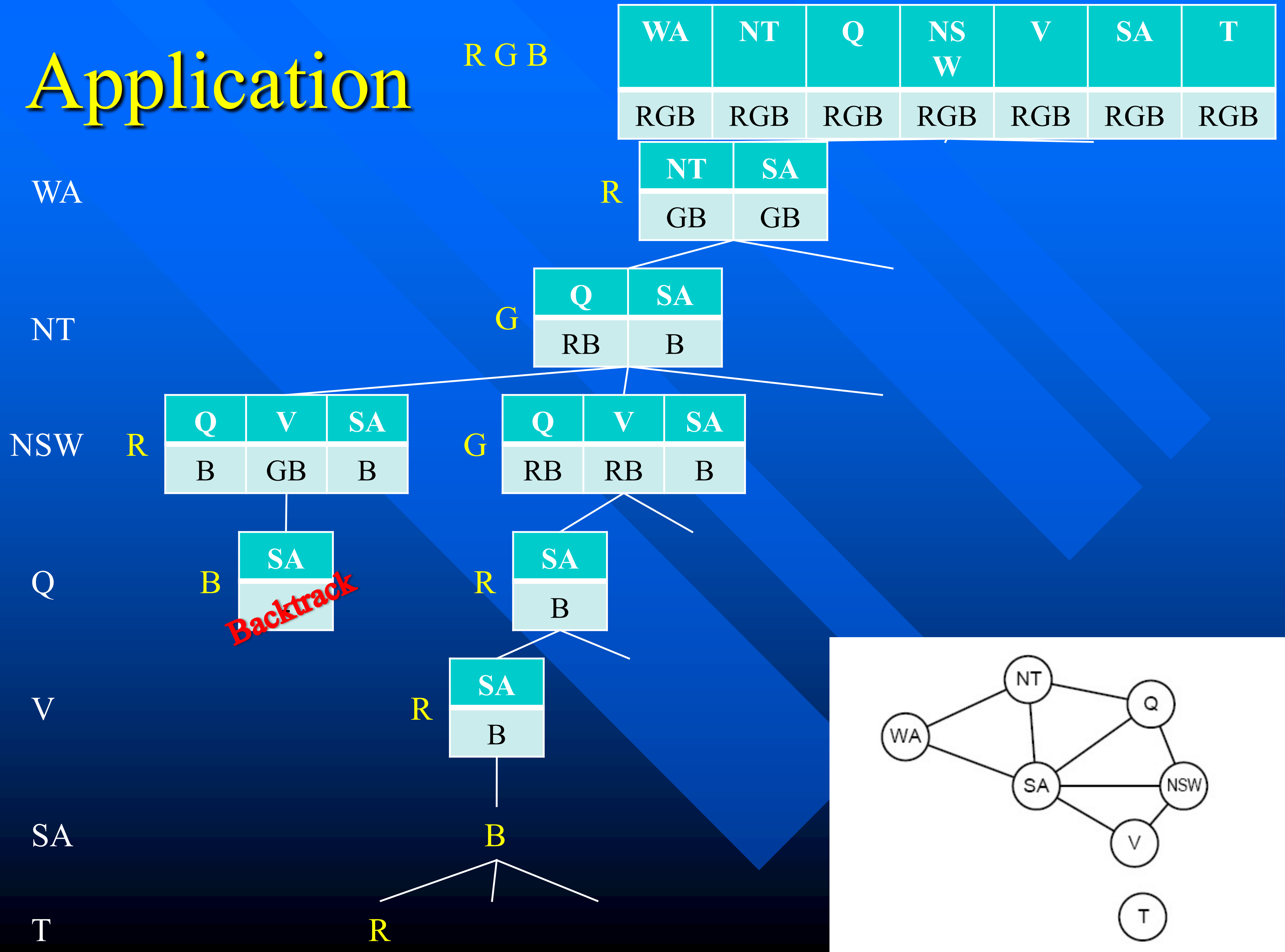
Forward Checking

```
Fonction FC(Assignation A, Network R) : Booléen
Début
    si |A| = |R.X| alors
            afficher a;
            retourner true;
    finsi;
    x ← ChoixVariableNonAssignée(R.X, A);
    D_old ← D ;
    pour tout v ∈ Tri(R.D(x)) faire
            si Propage(x, v, A) alors
                        si FC(A∪{(x,v)}, R) alors retourner true;
            finsi;
            D ← D_old ;
    finpour;
    retourner false;
Fin
```

Forward Checking

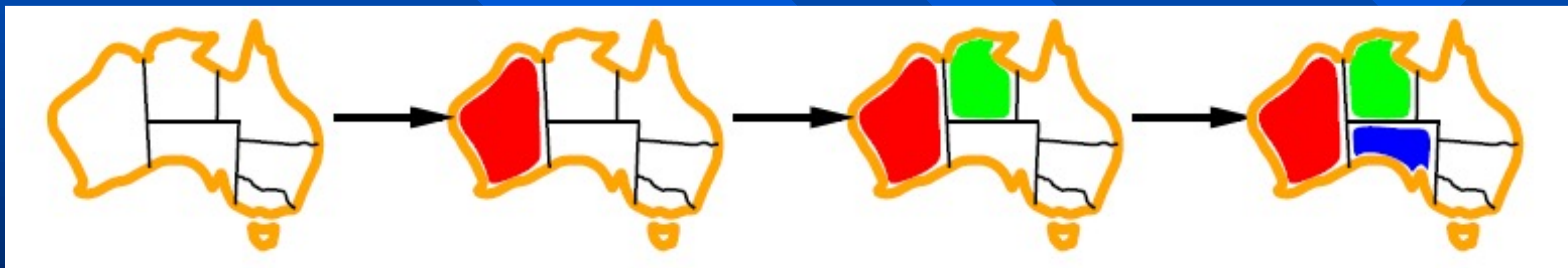
```
Fonction Propage(Variable x, Valeur v, Assignation A) : Booléen
Accès en modification : Network R
Début
  pour tout c ∈ R.C  tel que  x ∈ P(c)  et  |P(c)-(var(A)∪{x})|=1 faire
    pour tout w ∈ R.D(y) où y est l'unique variable de P(c)-(var(A)∪{x}) faire
      si il n'existe pas un tuple support pour w alors
        supprimer w de R.D(y) ;
      finsi ;
    finpour;
    si R.D(y)=∅ alors retourner false ;
  finpour ;
  retourner true ;
Fin ;
```

Application



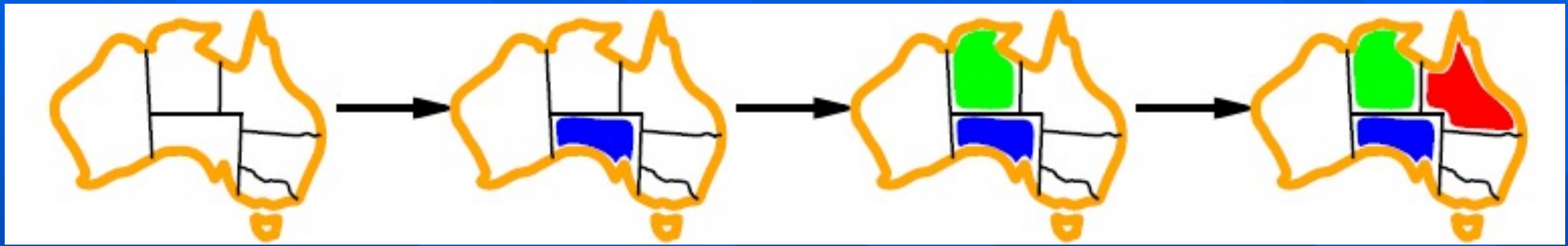
Les idées d'heuristiques pour les domaines dynamiques

- dom [Haralick & Eliott, 1980] : choix de la variable qui a le plus petit nombre de valeurs restantes dans son domaine courant



Les idées d'heuristiques pour les domaines dynamiques

- dom+deg : dom est en cas d'égalité deg (plus grand nombre de contraintes portant sur elle)



- dom/deg : choix de la variable qui a le plus petit ratio taille du domaine courant sur nombre de contraintes portant sur elle
- Variantes avec
  - ddeg : (dynamic) degré en enlevant du graphe les variables assignées
  - wdeg : (weighted) degré pondéré par le poids des contraintes où le poids est dynamiquement calculé en comptant le nombre de viols provoqué par les contraintes adjacentes