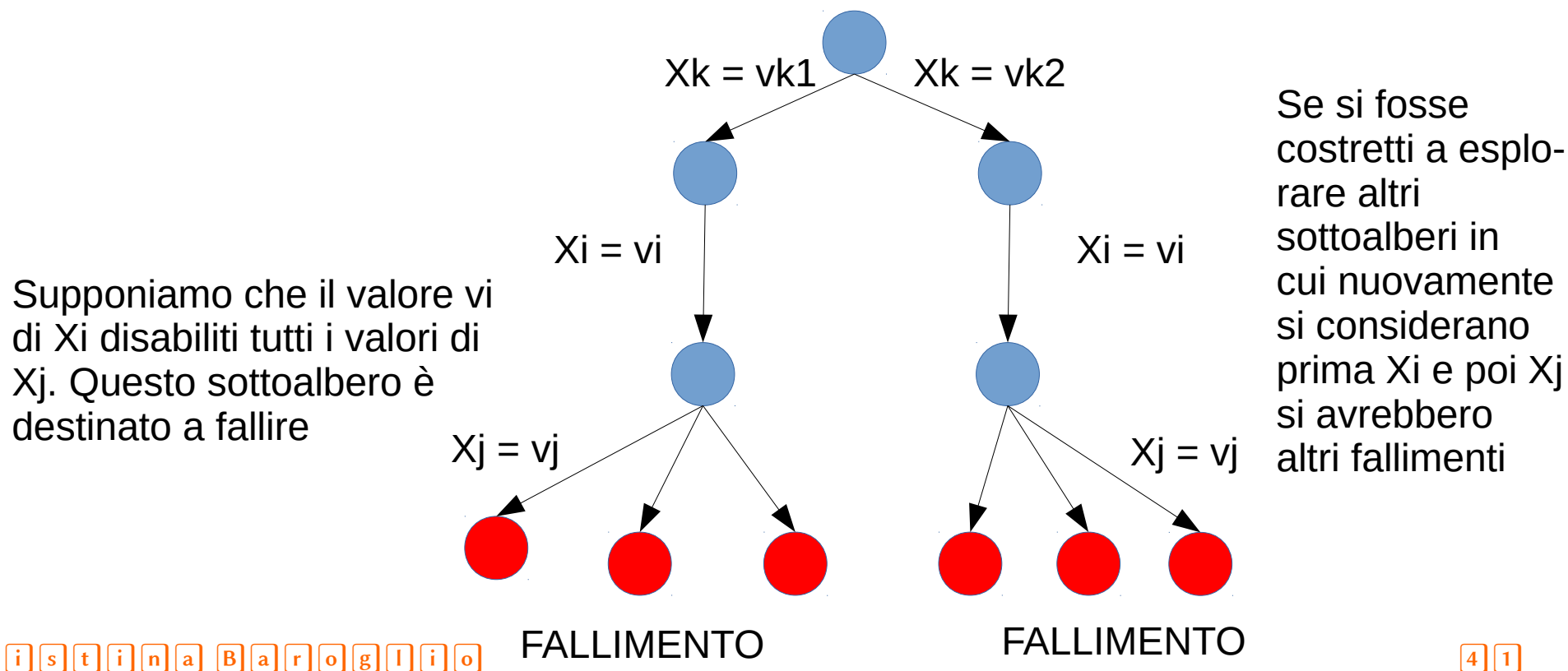


Ricerca con backtracking: osservazioni

- È una ricerca in profondità, non informata
- Non è particolarmente efficiente, in particolare è soggetta a thrashing (lo stesso assegnamento errato può essere ripetuto in più punti dell' albero), esempio:

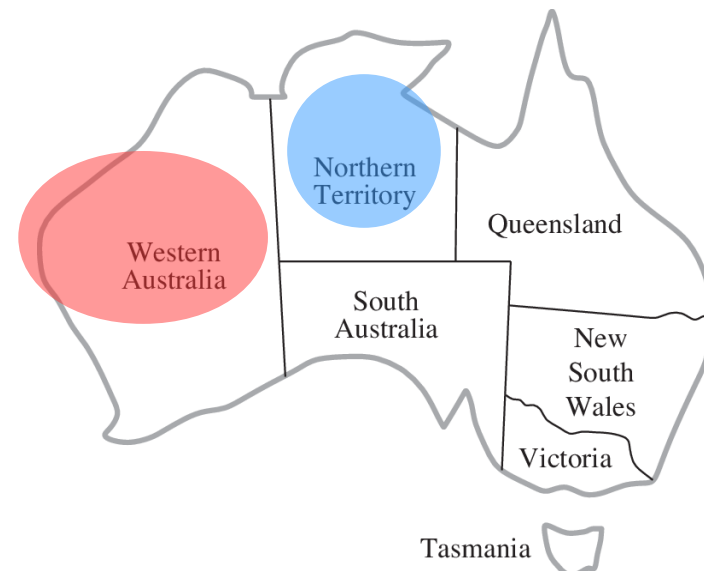


Ricerca con backtracking: osservazioni

- È una ricerca in profondità, non informata
- Non è particolarmente efficiente, in particolare è soggetta a thrashing (lo stesso assegnamento errato può essere ripetuto in più punti dell' albero)
- **Direzioni di miglioramento, richiedono conoscenza sui vincoli:**
 - 1) L' **assegnamento della variabile corrente** ha un impatto sulle variabili non ancora assegnate?
 - 2) L' **ordinamento dei valori** da provare influisce sul risultato?
 - 3) Quando un cammino fallisce, è possibile fare tesoro dell' esperienza per **evitare fallimenti simili** lungo altri cammini?

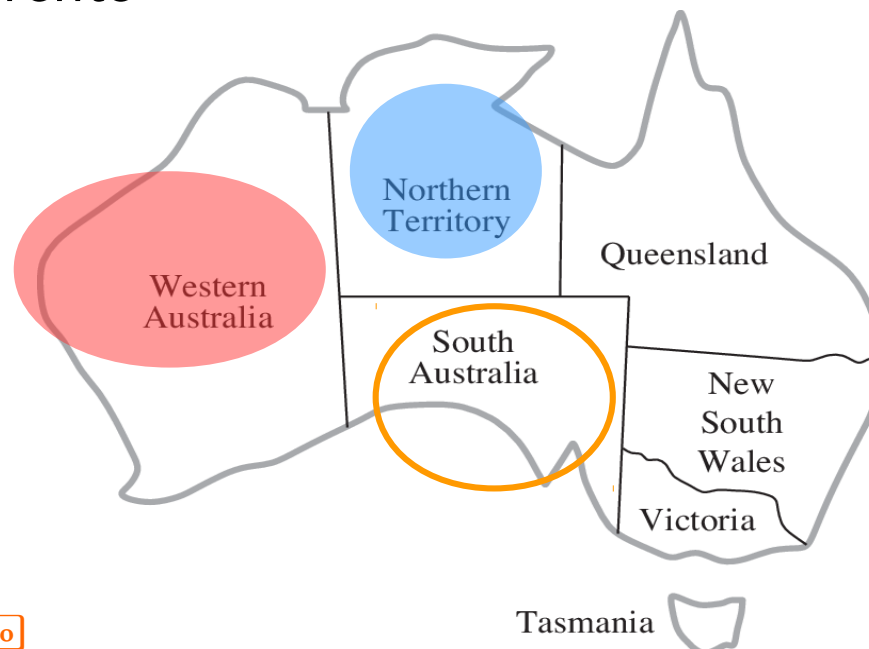
Euristiche di scelta della variabile

- L' algoritmo con backtracking sceglie, a ogni iterazione una variabile non assegnata:
 - *Se, invece di una scelta casuale, si adotta un' **euristica di scelta** si possono migliorare le prestazioni*
- **Esempio:** supponiamo si sia scelto il rosso per WA e il blu per NT, rimangono 5 variabili da assegnare
- Quale è più sensato scegliere?



Euristiche di scelta della variabile

- La più vincolata, cioè SA per la quale esiste uno solo valore possibile
- SA ha il vincolo di *non essere blu o rossa*, Q ha il vincolo di *non essere blu*, gli altri territori al momento potrebbero essere di qualsiasi colore
- **Euristica Minimum Remaining Values (o fail-first)**: sceglie una delle variabili con il minor numero di valori alternativi consistenti con l' assegnamento corrente



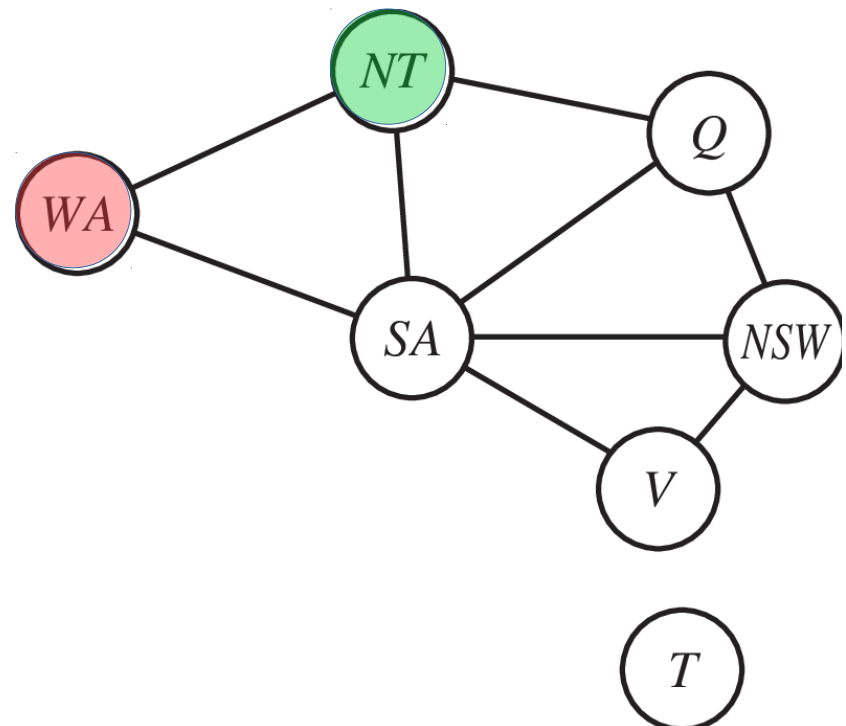
Euristiche di scelta della variabile

- *Quando MRV non è di aiuto (esempio, nella scelta della prima variabile da assegnare o in generale in caso di pareggio) si usa l' euristica di grado che sceglie la variabile coinvolta in più vincoli*
- nell' esempio, se l' assegnamento iniziale fosse solo $NT=B$:
 - WA, SA e Q sarebbero a parimerito per MRV con due valori consistenti
 - l' euristica di grado sceglierebbe SA che è coinvolta in 5 vincoli

Euristiche di scelta del valore

- *Scelta una variabile che può assumere diversi valori, quale valore considerare prima?*
- **Euristica del valore meno vincolante:** prediligere il valore che lascia più libertà alle variabili adiacenti sul grafo dei vincoli

Supponiamo di avere assegnato **WA=R** e **NT=G** e di dover assegnare **Q**



Euristica del valore meno vincolante

Valori possibili per Q: **B** e **R**

Valori possibili per i vicini di Q:

SA: **B**

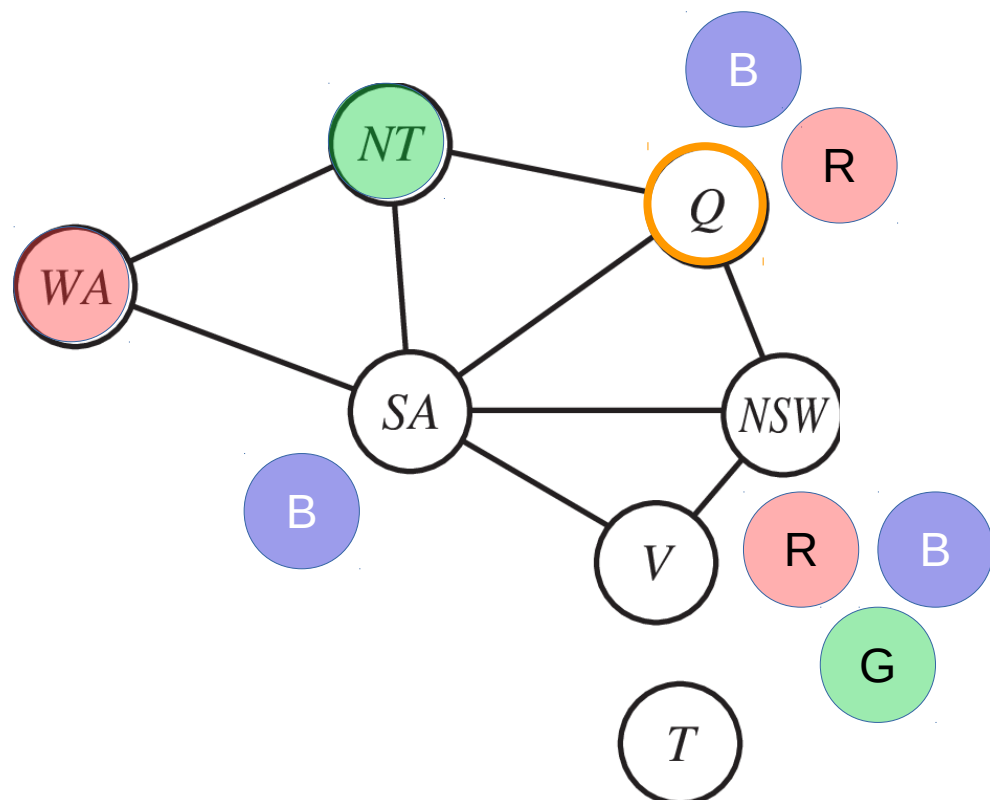
NSW: **R**, **G**, **B**

Se Q assume valore B: SA non può più assumere valori e i valori di NSW si riducono a 2

Se Q assume valore R: i valori possibili per SA non sono ridotti mentre NSW potrà essere G o B

È meno vincolante $Q=R$

Una scelta restrittiva anticipata rischia di tagliare via soluzioni



Rendiamo informata la ricerca

- La ricerca nello spazio dei possibili assegnamenti può essere migliorata usando meglio la conoscenza data dai vincoli
- Si alternano fasi di esplorazione a fasi di inferenza
- L' inferenza è espressa in termini di propagazione dei vincoli

Backtracking con inferenza

```
function RICERCA-BACKTRACKING(csp) returns una soluzione, o il fallimento  
  return BACKTRACKING({ }, csp)
```

```
function BACKTRACKING(assegnamento, csp) returns una soluzione, o il fallimento  
  if assegnamento è completo then return assegnamento  
  var ← SCEGLI-VARIABILE-NON-ASSEGNATA(csp)  
  for each valore in ORDINA-VALORI-DOMINIO(var, assegnamento, csp) do  
    if valore è consistente con assegnamento then  
      aggiungi {var = valore} ad assegnamento  
      inferenze ← INFERENZA(csp, var, valore)  
      if inferenze ≠ fallimento then  
        aggiungi inferenze ad assegnamento  
        risultato ← BACKTRACKING(assegnamento, csp)  
        if risultato ≠ fallimento then  
          return risultato  
      rimuovi {var = valore} e inferenze da assegnamento  
  return fallimento
```

Figura 6.5 Un semplice algoritmo di backtracking per problemi di soddisfacimento di vincoli. L'algoritmo

Consistenza locale

- **Scopo:** *ridurre lo spazio di ricerca in modo consistente usando i vincoli per tagliare assegnamenti non consistenti con l' informazione disponibile*
- **Metodo di consistenza locale:** focalizzarsi su una **proprietà di consistenza relativa ad una parte del CSP** (un sottoproblema) e poi propagare questa informazione in tutto il CSP cercando di costruire una soluzione
- Fra le proprietà considerate:
 - 1) **Forward checking** (è un approccio più che una proprietà)
 - 2) **Node consistency:** riguarda singole variabili, vale quando i vincoli unari sono soddisfatti da tutti i valori dei domini delle rispettive variabili
 - 3) **Arc consistency:** proprietà direzionale relativa a un vincolo binario (vediamo più avanti), tutti i valori consistenti di una variabile x possono essere estesi a y tramite i vincoli
 - 4) **Path consistency:** proprietà che lega una coppia di variabili a una terza tramite i vincoli (vedremo più avanti)

Forward checking

- **Osservazione:** se $WA=R$ a causa dei vincoli le confinanti NT e SA non potranno assumere questo valore
- **Forward checking:**
si percorrono gli archi che collegano il nodo, corrispondente alla variabile assegnata, con i suoi vicini diretti e si riduce il range dei possibili valori di tali vicini in maniera conforme al vincolo
- **Esempio:**
quando WA diventa R si toglie R dai valori possibili di NT e SA
- Il forward checking si accompagna naturalmente con l'euristica MRV

Esempio

- La tabella riporta riduzioni progressive dei domini delle variabili. Vengono propagate dai vincoli attraverso al grafo dei vincoli a seguito degli assegnamenti provati. Quando una variabile (SA) non ha più valori legali si ha il fallimento e di conseguenza il backtracking

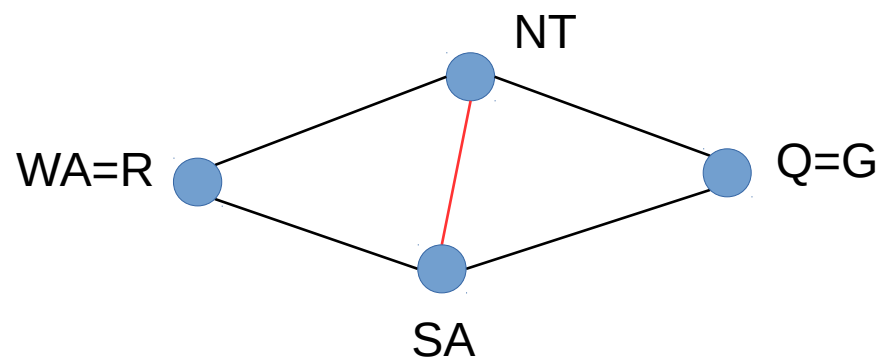
	WA	NT	Q	NSW	V	SA	T
INIZIO	R G B	R G B	R G B	R G B	R G B	R G B	R G B
WA=R →	R	G B	R G B	R G B	R G B	G B	R G B
Q=G →	R	B	G	R B	R G B	B	R G B
V=B →	R	B	G	R	B		R G B

Confinano tutti con SA

Limiti del forward checking

- Il forward checking non cattura tutte le inconsistenze

WA	NT	Q	NSW	V	SA	T
R G B	R G B	R G B	R G B	R G B	R G B	R G B
R	G B	R G B	R G B	R G B	G B	R G B
R	B	G	R B	R G B	B	R G B
R	B	G	R	B		R G B



Questo assegnamento forza un'inconsistenza infatti NT e SA, confinanti sono richiesti avere lo stesso valore!!

Forward checking non se ne accorge

Oltre il forward checking

- Node consistency
- Arc consistency
- Path consistency

Node consistency

- Un grafo di vincoli è **node consistent** quando lo sono tutte le sue variabili
- Una variabile è **node consistent** quando rispetta tutti i vincoli unari che la riguardano
- **Esempio:** “età compresa fra 18 e 35 anni” può essere espresso con due vincoli unari: $\text{età} > 18$ e $\text{età} < 35$
- L' algoritmo di node consistency controlla che un grafo di vincoli sia node consistent valutando tutte le variabili e i loro domini correnti rispetto ai vincoli unari

Arc consistency

- Un grafo di vincoli è **arc consistent** quando rispetta tutti i vincoli binari
- È una proprietà importante perché: dato un vincolo a tre o più variabili è sempre possibile trasformarlo in un insieme di vincoli binari
- **NB:** il forward checking realizza un controllo di arc consistency per i vincoli che legano una variabile, alla quale è stato assegnato un valore, e i suoi vicini diretti sul grafo dei vincoli ma non propaga il ragionamento sui vicini dei vicini, ecc.

Arc consistency

- Dato un grafo di vincoli si intende per **arco** un *lato orientato*
- **Esempio:** $WA \rightarrow NSW$ sarà diverso da $NSW \rightarrow WA$
- Un arco è **consistente** quando:
per ogni valore possibile della variabile corrispondente al vertice sorgente,
esiste un qualche valore del vertice destinazione che è ad esso consistente
- **Esempio:**
 - se il dominio di **WA** è $\{R, G\}$ e il dominio di **NSW** è $\{R\}$
 - $WA \rightarrow NSW$ non è consistente
se WA assume valore R, NSW non ha valori assegnabili
 - $NSW \rightarrow WA$ è consistente
- Si può cercare di rendere consistenti gli archi che non lo sono cancellando valori dai loro domini