

## 2.4 Jeux de paquets et jeux de Böchi

CHARPENTIER  
Clément (2)

Comment construire  $G'$  ?

On procède en  $k$  étapes,  $k$  étant le nombre de couleurs possibles pour les sommets de  $G$ .

### Étape 1

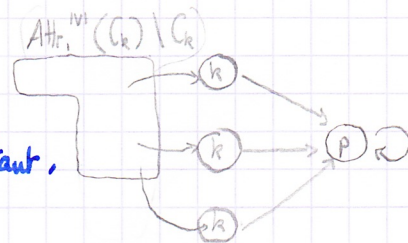
On sait qu'atteindre un sommet de couleur  $k$  désigne le gagnant ( $J_0$  si  $k$  pair,  $J_1$  si  $k$  impair).

On calcule  $\text{Attr}_0^{IV}(C_k)$  si  $k$  pair, ou  $\text{Attr}_1^{IV}(C_k)$  si  $k$  impair.

( $C_k$  est l'ensemble des sommets de couleur  $k$ ).

- Si  $k$  impair :  $\text{Attr}_1^{IV}(C_k) \subseteq \text{Win}_1$ .

Pour que cela soit le cas dans  $G'$ , on crée un état-puit  $p$  non acceptant, et une transition vers  $p$  depuis chaque sommet de couleur  $k$ .



- Si  $k$  pair :  $\text{Attr}_0^{IV}(C_k) \subseteq \text{Win}_0$ .

Pour que cela soit le cas dans  $G'$ , on crée une boucle à chaque état de couleur  $k$  à l'état par défaut, et on attribue les sommets de couleur  $k$  de  $v_1$  à  $v_0$ . Enfin, on déclare ces sommets acceptants dans  $G'$ .



### Étapes 2 ... k

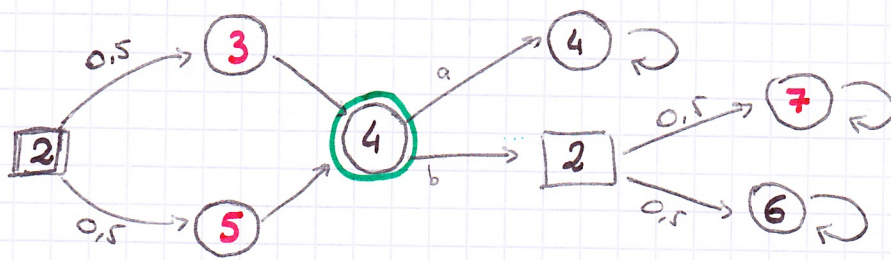
À l'étape 2, on s'occupe de la couleur  $k-1$ , à l'étape 3, de  $k-2$  ... jusqu'à 0 à l'étape  $k$ .

Soit  $i$  la couleur de l'étape actuelle. On s'intéresse à  $G \setminus \sum_{i=1}^k$



- On voit bien que ces consignes-là sont positionnelles.  
La stratégie héritée du jeu de Büchi étant également positionnelle, tous les sommets de  $G$  ont des consignes positionnelles.
- Question subsidiaire : non, un jeu stochastique de partie faible n'a pas toujours une stratégie positionnelle comme stratégie optimale.

Exemple :



Si, en arrivant dans l'état vert, la partie est passée par l'état de couleur 3, alors la stratégie optimale serait de prendre le choix a.

Si la partie est passée par la couleur 5, la stratégie optimale serait de choisir b.

### 3.1 Jeux de parité sur des arêtes aux grands cycles

On utilise deux choses pour cet algorithme :  
- une file de sommets  $F$   
- un étiquetage booléen  $Win$  sur les sommets :  
 $Win(s) = 0 : s \in Win_0$   
 $Win(s) = 1 : s \in Win_1$   
par défaut,  $Win(s) \neq 0$   
 $\neq 1$   
 $= NULL$ ;

```
Pour tout  $v \in V$  faire {  
  Si  $v$  est un sommet pair alors {  
    si  $\chi(v)$  paire alors  
       $Win(v) = 0$ ;  
    sinon  
       $Win(v) = 1$ ;  
  }  
  enfila( $F, v$ );  
}
```

Tant que non-vide ( $F$ ) faire {

$v = \text{defila}(F)$ ;  
Si  $\text{succ}(v) \neq \emptyset$  alors {

```
  Pour tout  $x \in \text{succ}(v)$  faire {  
    si ( $Win(x) = NULL$ ) alors {  
      si ( $x \in V_0 \wedge Win(v) = 0$ )  
         $Win(x) = 0$ ;  
      sinon { si ( $x \in V_1 \wedge Win(v) = 1$ )  
         $Win(x) = 1$ ;  
      }  
    }
```

```
    si pour tout  $w \in \text{succ}(v), Win(w) = 0$  alors  
       $Win(v) = 0$ ;  
    sinon {
```

```
      si pour tout  $w \in \text{succ}(v), Win(w) = 1$  alors  
         $Win(v) = 1$ ;  
      }  
    }
```

```
  }  
  si ( $Win(x) \neq NULL$ ) alors  
    empila( $F, v$ );  
  sinon
```

```
    empila( $F, x$ );  
  }
```

*\* l'algorithme est encore très mal optimisé \**

Il est néanmoins polynomial : il passe au maximum une fois sur chaque sommet.