

## 2.4 Jeux de puits et jeux de Büchi

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'une partie atteignant un sommet de couleur  $k$  a comme gagnant  $J_0$  si  $k$  pair,  $J_1$  si  $k$  impair.

Soit  $C_k$  l'ensemble des sommets de couleur  $k$ .

— Si  $k$  impair : On calcule  $\text{Attr}_1^{VI}(C_k) \subseteq \text{Win}_1$ .

Pour que ce soit également le cas dans  $G'$ , on crée un état-puits  $p$ , et une transition vers  $p$  depuis chaque sommet de couleur  $k$ .

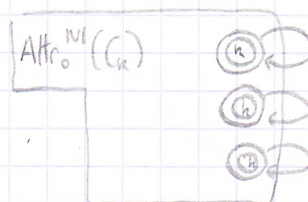
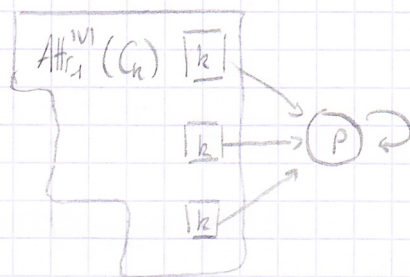
Dans le jeu de Büchi  $G'$ ,  $p$  sera non-acceptant.

Enfin, on attribue tous les sommets de couleur  $k$  à  $V_1$ .

— Si  $k$  pair : On calcule  $\text{Attr}_0^{VI}(C_k) \subseteq \text{Win}_0$ .

Pour que cela soit également le cas dans  $G'$ , on crée une boucle à chaque état de couleur  $k$  n'en étant pas doté.

On déclare ces états comme acceptants dans  $G'$ , et on les attribue à  $V_0$ .

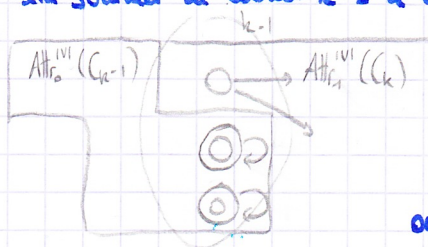


### Étape 2

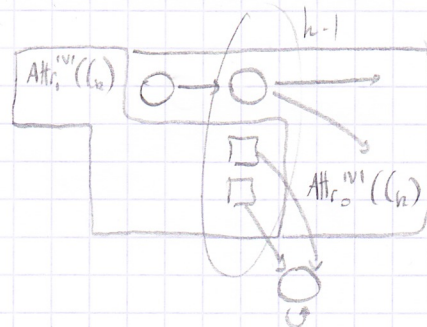
Dans  $V \setminus \{\text{Attr}_0^{VI}(C_k)\}$  (ou  $V \setminus \{\text{Attr}_1^{VI}(C_k)\}$ ) :

On sait qu'une partie atteignant un sommet de couleur  $k-1$  a comme gagnant  $J_0$  si  $k-1$  pair,  $J_1$  si  $k-1$  impair.

On fait donc de même :



ou :



Étapes 3 ... k : On fait de même jusqu'à 0.

Dans  $G'$  ainsi construit : un sommet de  $V$  gagnant dans  $G$  ( $\Rightarrow$ ) ce sommet est gagnant dans  $G'$ .

La stratégie de  $J_0$  dans  $G$  se déduit de celle dans  $G'$  : elle est la même dans tous les sommets sauf ceux pointant vers  $p$  et ceux, acceptants, qui bouclent.

Pour ces sommets, là, la stratégie consiste à éviter l'attraction d'une couleur supérieure (possible, sinon ce sommet serait dans l'attraction de la couleur supérieure en question).



### 3.1 Jeux de paillie sur des arènes sans 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  $2(v)$  pair alors
       $Win(v) = 0$ ;
    sinon
       $Win(v) = 1$ ;
  }
  empiler( $F, v$ );
}

```

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

$v = \text{defiler}(F)$ ;

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  $(x \in V_1 \wedge Win(v) = 1)$

$Win(x) = 1$ ;

    sinon {

      si pour tout  $w \in \text{succ}(x)$ ,  $Win(w) = 0$  alors

$Win(x) = 0$ ;

      sinon {

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

$Win(x) = 1$ ;

      }};

    } si  $Win(x) = NULL$  alors

      empiler( $F, v$ );

  sinon

    empiler( $F, x$ );

}

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

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