

3.3 Inclure des notions d'équité

Exécutions équitables

- Chaque processus est activé infiniment souvent : $\bigwedge_i (GF \text{ enabled}_i)$
- Aucun processus ne reste infiniment dans la section critique : $\bigwedge_i \neg(FG \text{ critic}_i) = \bigwedge_i GF(\neg \text{critic}_i)$

Contraintes d'équité

- Contrainte d'équité inconditionnelle : $GF\varphi$
- Contrainte d'équité forte : $GF\varphi \rightarrow GF\varphi'$
- Contrainte d'équité faible : $FG\varphi \rightarrow GF\varphi'$

Conditions d'équité

- Une condition d'équité est une conjonction de contraintes d'équité
- Une condition d'équité est une formule LTL!

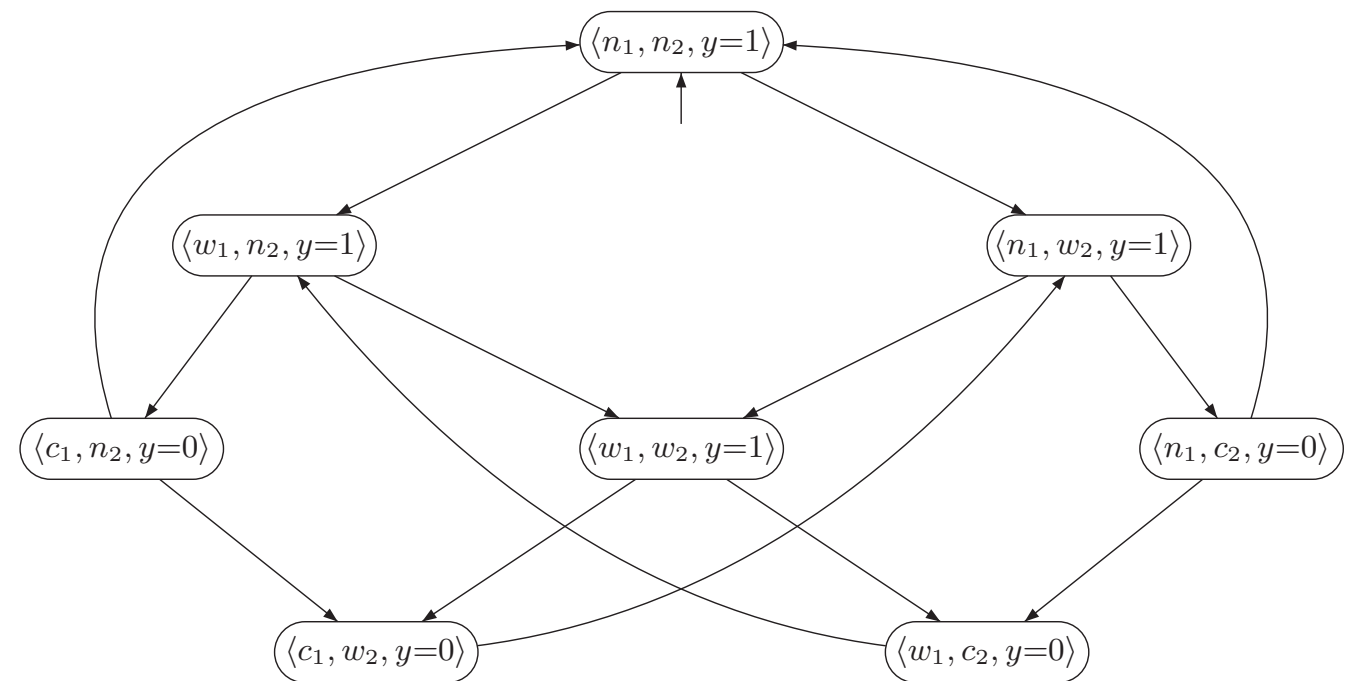
Exécutions équitables

- Soit t une trace d'exécution d'une structure de Kripke M , *fair* une condition d'équité
- t est *équitable* si $t, 0 \models \text{fair}$

LTL équitable

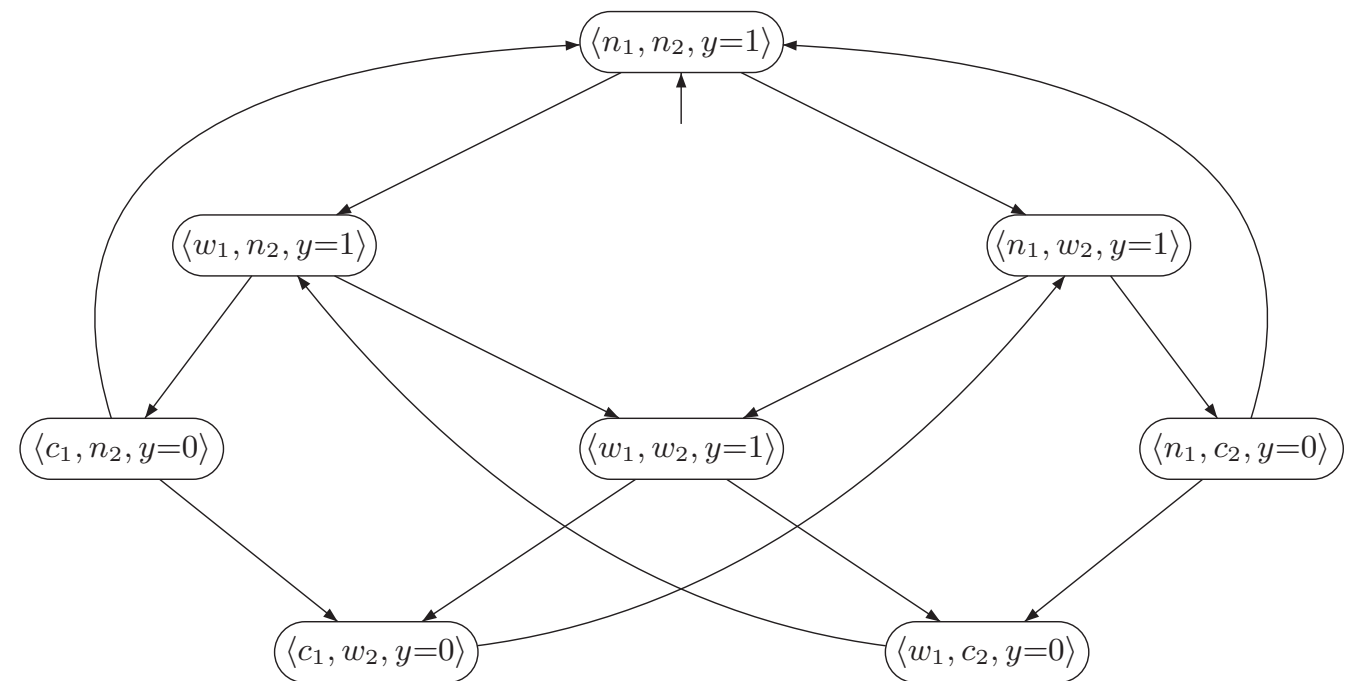
- Soit une structure de Kripke M , *fair* une condition d'équité et φ une formule LTL.
- $M \models_{\text{fair}} \varphi$ ssi $t, 0 \models_{\text{fair}} \varphi$ pour toute trace initiale t de M ssi $t, 0 \models \varphi$ pour toute trace initiale équitable de M .

Example



Example

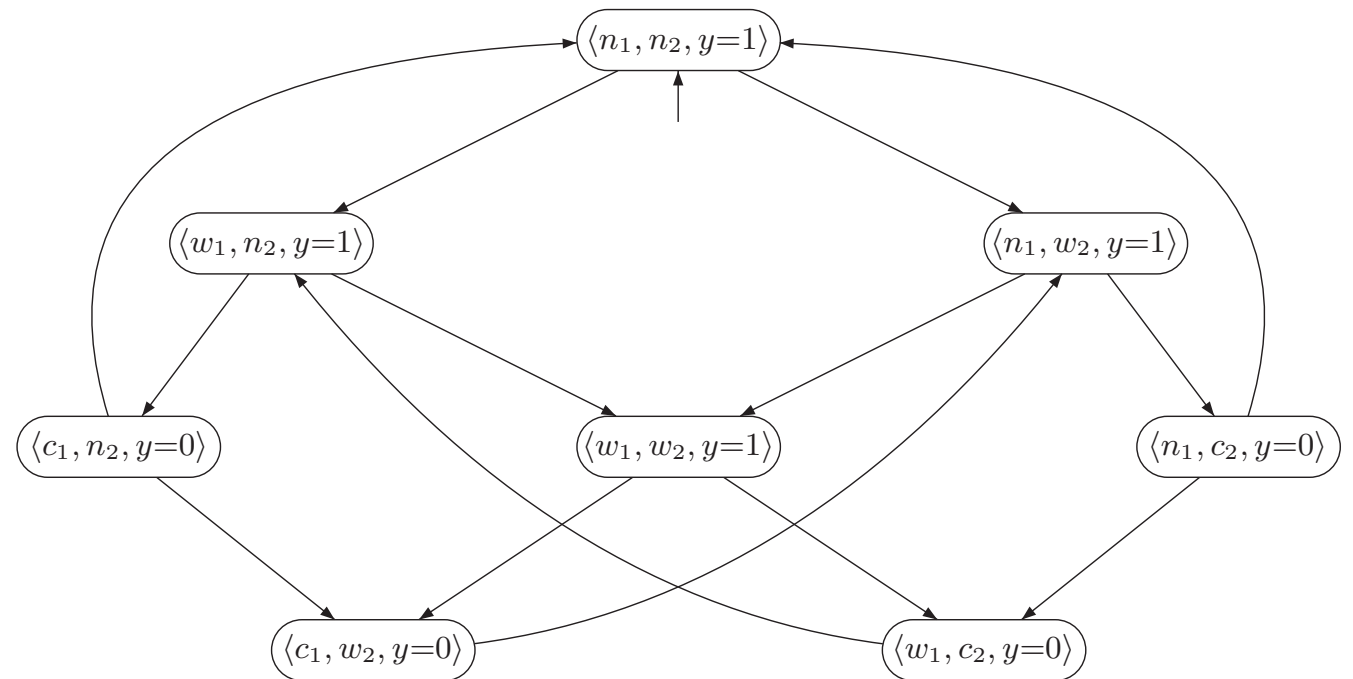
$$GF(w_1 \wedge \neg c_2) \rightarrow GFc_1 \wedge GF(w_2 \wedge \neg c_1) \rightarrow GFc_2$$



Example

$$GF(w_1 \wedge \neg c_2) \rightarrow GFc_1 \wedge GF(w_2 \wedge \neg c_1) \rightarrow GFc_2$$

\wedge

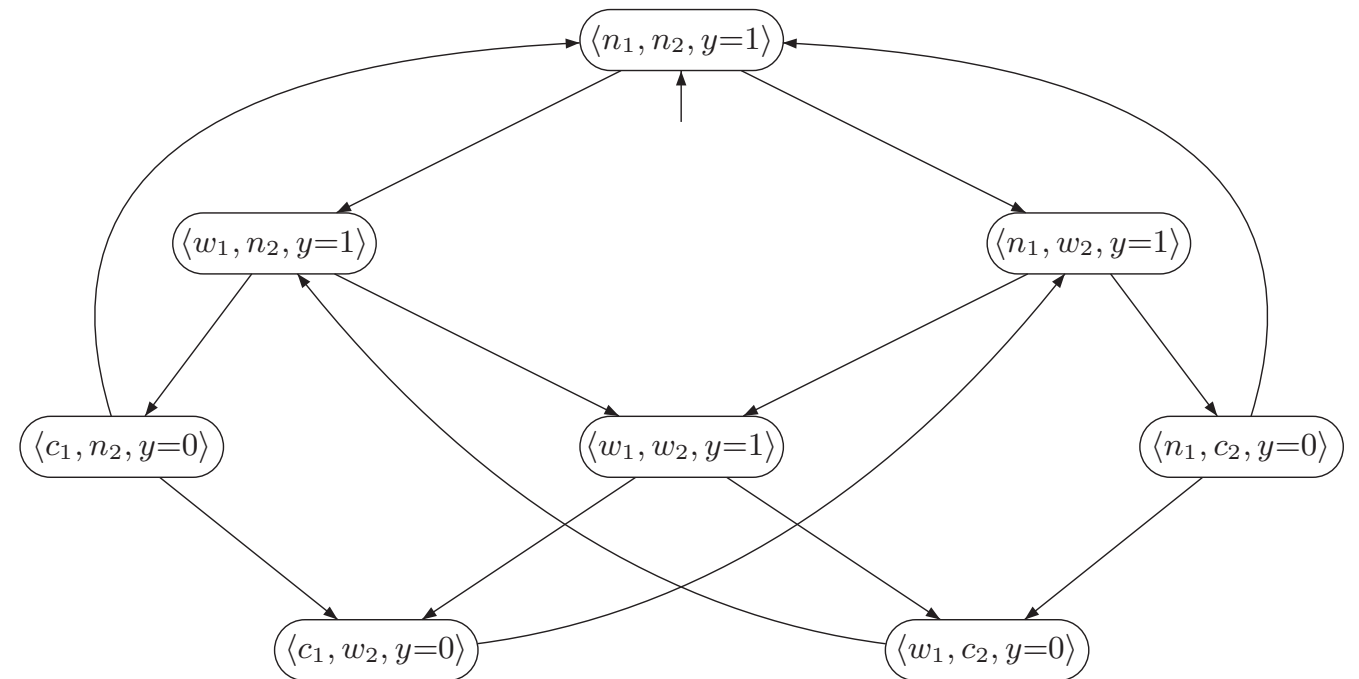


Example

$$GF(w_1 \wedge \neg c_2) \rightarrow GFc_1 \wedge GF(w_2 \wedge \neg c_1) \rightarrow GFc_2$$

\wedge

$$(FGn_1 \rightarrow GFw_1) \wedge (FGn_2 \rightarrow GFw_2)$$



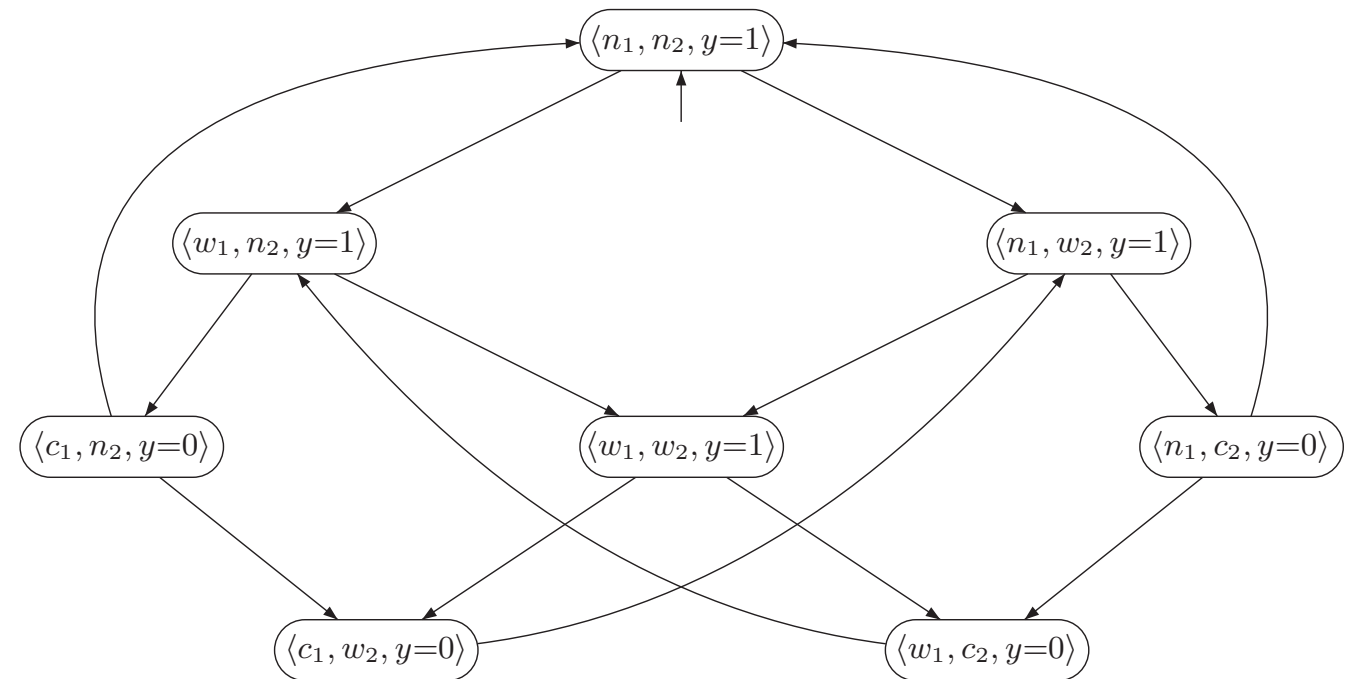
Example

$$GF(w_1 \wedge \neg c_2) \rightarrow GFc_1 \wedge GF(w_2 \wedge \neg c_1) \rightarrow GFc_2$$

\wedge

$$(FGn_1 \rightarrow GFw_1) \wedge (FGn_2 \rightarrow GFw_2)$$

$$M \models_{fair} GFc_1 \wedge GFc_2$$



Model-Checking LTL équitable

Théorème : $M \models_{fair} \varphi$ ssi $M \models fair \rightarrow \varphi$.

CTL équitable

- Conditions d'équité ne peuvent pas s'écrire en CTL
- On voudrait dire $A(\textit{fair} \rightarrow \varphi)$ ou $E(\textit{fair} \wedge \varphi)$ mais ce sont des formules CTL*

CTL équitable

$$\varphi ::= p \in AP \mid \neg \varphi \mid \varphi \vee \varphi$$
$$\mid E_f X \varphi \mid A_f X \varphi \mid E_f \varphi U \varphi \mid A_f \varphi U \varphi$$

$s \models p$ ssi $p \in I(s)$

$s \models \neg \varphi$ ssi $s \not\models \varphi$

$s \models \varphi_1 \vee \varphi_2$ ssi $s \models \varphi_1$ ou $s \models \varphi_2$

$s \models E_f X \varphi$ ssi il existe une exécution **équitable** $s_0 s_1 \dots$ tel que $s_0 = s$, t.q. $s_1 \models \varphi$

$s \models A_f X \varphi$ ssi s', pour toute exécution **équitable** $s_0 s_1 \dots$ telle que $s_0 = s$, $s_1 \models \varphi$

$s \models E_f \varphi_1 U \varphi_2$ ssi il existe une exécution **équitable** $s_0 s_1 \dots s_k$ tel que $s_0 = s$, $s_k \models \varphi_2$ et pour tout $0 \leq i \leq k$, $s_i \models \varphi_1$.

$s \models A_f \varphi_1 U \varphi_2$ ssi pour toute exécution **équitable** $s_0 s_1 \dots$ telle que $s_0 = s$, il existe k t.q. $s_k \models \varphi_2$ et pour tout $0 \leq i \leq k$, $s_i \models \varphi_1$.

Model-Checking de CTL équitale

- On suppose qu'on a étiqueté les états avec une nouvelle AP **fair**, qui indique s'il existe une exécution équitale partant de l'état
- $s \models E_f X \varphi$ ssi $s \models EX(\varphi \wedge \text{fair})$
- $s \models A_f X \varphi$ ssi $s \models AX(\neg \text{fair} \vee \varphi)$
- $s \models E_f \varphi U \varphi'$ ssi $s \models E \varphi U (\text{fair} \wedge \varphi')$
- $s \models A_f \varphi U \varphi'$ ssi $s \models \neg E_f G \neg \varphi' \wedge \neg E_f (\neg \varphi' U (\neg \varphi \wedge \neg \varphi'))$
ssi $s \models \neg E_f G \neg \varphi' \wedge \neg E (\neg \varphi' U (\text{fair} \wedge \neg \varphi \wedge \neg \varphi'))$

Model-Checking de CTL équitable

- 1er problème : Comment calculer fair?
- Rappel : $s \models \text{fair}$ ssi il existe une exécution équitable partant de s
- \rightarrow Dépend de la condition d'équité!
- 2ème problème : calculer $E_f G \varphi$

Calculer fair : les composantes fortement connexes

Définition : Dans un graphe, une composante fortement connexe (SCC) est un sous-graphe maximal tel que pour toute paire de noeuds (s, s') s' est accessible depuis s , et s est accessible depuis s'

L'algorithme de **Tarjan** permet de calculer les SCC d'un graphe en temps linéaire.

Calculer fair : le cas inconditionnel

- On considère une condition d'équité de la forme $\bigwedge_i GF\varphi_i$, avec φ_i formule CTL.
- On marque les états par les φ_i .
- On calcule les SCC de M par l'algorithme de Tarjan.
- Soit S' l'union des SCC qui intersectent $S(\varphi_i)$, pour tout i .
- fair est l'ensemble des états pouvant atteindre S' .
- (accessibilité se calcule en temps linéaire)

Calculer $E_f G \varphi$: le cas inconditionnel

- Effectuer $\text{mark}(\varphi)$.
- Soit $M(\varphi)$ la restriction de M aux états de $S(\varphi)$.
- Calculer les SCC de $M(\varphi)$ (algo de Tarjan).
- Soit S' l'union de SCC de $M(\varphi)$ intersectant $S(\varphi_i)$, pour tout i .
- $M, s \models E_f G \varphi$ ssi $M, s \models E \varphi U S'$ ssi $M(\varphi), s \models EFS'$.
- \rightarrow problème d'accessibilité.

Model-Checking de CTL équitale

- On suppose qu'on a étiqueté les états avec une nouvelle AP **fair**, qui indique s'il existe une exécution équitale partant de l'état
- $s \models E_f X \varphi$ ssi $s \models EX(\varphi \wedge \text{fair})$
- $s \models A_f X \varphi$ ssi $s \models AX(\neg \text{fair} \vee \varphi)$
- $s \models E_f \varphi U \varphi'$ ssi $s \models E \varphi U (\text{fair} \wedge \varphi')$
- $s \models A_f \varphi U \varphi'$ ssi $s \models \neg E_f G \neg \varphi' \wedge \neg E_f (\neg \varphi' U (\neg \varphi \wedge \neg \varphi'))$
ssi $s \models \neg E_f G \neg \varphi' \wedge \neg E (\neg \varphi' U (\text{fair} \wedge \neg \varphi \wedge \neg \varphi'))$

Model-Checking de CTL équitable

- On suppose qu'on a étiqueté les états avec une nouvelle AP **fair**, qui indique s'il existe une exécution équitable partant de l'état ✓
- $s \models E_f X \varphi$ ssi $s \models EX(\varphi \wedge \text{fair})$
- $s \models A_f X \varphi$ ssi $s \models AX(\neg \text{fair} \vee \varphi)$
- $s \models E_f \varphi U \varphi'$ ssi $s \models E \varphi U (\text{fair} \wedge \varphi')$
- $s \models A_f \varphi U \varphi'$ ssi $s \models \neg E_f G \neg \varphi' \wedge \neg E_f (\neg \varphi' U (\neg \varphi \wedge \neg \varphi'))$
ssi $s \models \neg E_f G \neg \varphi' \wedge \neg E (\neg \varphi' U (\text{fair} \wedge \neg \varphi \wedge \neg \varphi'))$

Model-Checking de CTL équitale

- On suppose qu'on a étiqueté les états avec une nouvelle AP **fair**, qui indique s'il existe une exécution équitale partant de l'état ✓
- $s \models E_f X \varphi$ ssi $s \models EX(\varphi \wedge \text{fair})$
- $s \models A_f X \varphi$ ssi $s \models AX(\neg \text{fair} \vee \varphi)$
- $s \models E_f \varphi U \varphi'$ ssi $s \models E \varphi U (\text{fair} \wedge \varphi')$
- $s \models A_f \varphi U \varphi'$ ssi $s \models \neg E_f G \neg \varphi' \wedge \neg E_f (\neg \varphi' U (\neg \varphi \wedge \neg \varphi'))$
ssi $s \models \neg E_f G \neg \varphi' \wedge \neg E (\neg \varphi' U (\text{fair} \wedge \neg \varphi \wedge \neg \varphi'))$ ✓

Et aussi...

- Autres logiques temporelles : CTL*, mu-calcul... (plus expressives), ForSpec, PSL, Sugar... (industrie)
- Méthodes efficaces : méthodes symboliques, techniques de réduction (ordres partiels...)