

Application de CLASS/UNION : test d'équivalence de deux automates déterministes

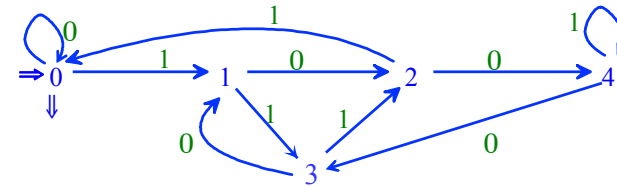
501

Automates déterministes

 $a = (Q, A, i, T, \delta)$

Q états
 A alphabet
 i état initial
 T états terminaux
 δ fonction de transition

ensemble fini
 ensemble fini
 $i \in Q$
 $T \subseteq Q$
 $\delta : Q \times A \rightarrow Q$


 $L(a) = \{x \in A^* / x \text{ étiquette d'un chemin de } i \text{ à } t \in T\}$
 $L = \{ \text{écritures en base 2 des multiples de 5} \}$

502

Algorithmes sur automates

- émondage
- calcul de $L(a)$
- expression rationnelle \rightarrow automate
- déterminisation
- minimisation
- tests « $L(a) = \emptyset ?$ » ou « $L(a) = A^* ?$ »
- Construction d'un automate a tel que
 - $L(a) = A^* - L(B)$
 - $L(a) = L(B) \cup L(C)$
 - $L(a) = L(B) \cap L(C)$
 - $L(a) = L(B)^*$
 - ...
- tester $L(a) = B$
- test s'équivalence : $L(a) = L(B) ?$
- ...

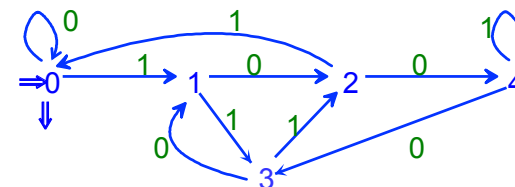
503

Représentation des automates

Représentation de la fonction de transition δ par

- listes des flèches
- matrice d'adjacence
- listes des successeurs
- table de transition

δ	0	1
0	0	1
1	2	3
2	4	0
3	1	2
4	3	1

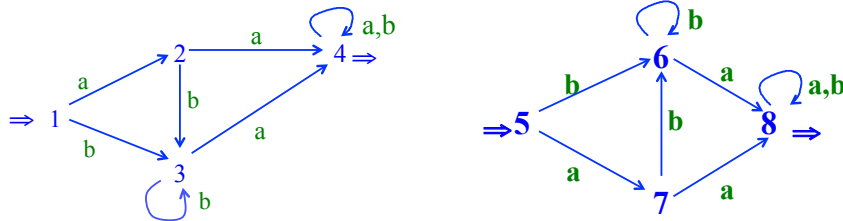


504

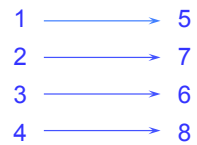
Égalité d'automates

UMLV ©

Tester l'égalité des structures



en construisant la bijection



505

Calcul de la bijection

UMLV ©

$$a_1 = (Q_1, A, i_1, T_1, \delta_1) \quad a_2 = (Q_2, A, i_2, T_2, \delta_2)$$

Calcul de la bijection f entre Q_1 et Q_2 par la procédure bij
 $\text{bij}(p, q)$ met en bijection p et q

début

$\text{bij}(i_1, i_2)$;

si les états sont en bijection par f **et**

pour tout $q_1, q_1 \in T_1$ ssi $f(q_1) \in T_2$ **alors**

« a_1 et a_2 sont égaux » ;

fin

506

Calcul de la bijection

UMLV ©

$$a_1 = (Q_1, A, i_1, T_1, \delta_1) \quad a_2 = (Q_2, A, i_2, T_2, \delta_2)$$

Test de bijection par calcul simultané de la fonction inverse : $g^{-1} = f$

procédure $\text{bij}(q_1, q_2)$;

début

si (un seul de $f(q_1)$ et $g(q_2)$ défini)

ou ($f(q_1)$ et $g(q_2)$ définis **et** $f(q_1) \neq q_2$) **alors**

« a_1 et a_2 non égaux » ;

sinon si $f(q_1)$ et $g(q_2)$ non définis **alors** {

$f(q_1) \leftarrow q_2$; $g(q_2) \leftarrow q_1$;

pour chaque $a \in A$ **faire**

$\text{bij}(\delta_1(q_1, a), \delta_2(q_2, a))$;

}

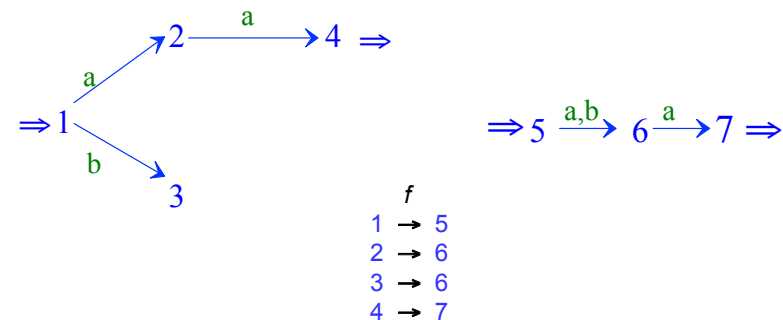
fin

507

Temps d'exécution : $O(\text{card}A \cdot \min\{\text{card}Q_1, \text{card}Q_2\})$

s'étend à des automates non complets

l'algorithme précédent ne peut se passer de g !



508

Équivalence d'automates

UMLV ©

Tester $L(\mathbf{a}_1) = L(\mathbf{a}_2)$ $n = \text{card } Q_1 + \text{card } Q_2$

- si \mathbf{a}_1 et \mathbf{a}_2 minimaux
 $L(\mathbf{a}_1) = L(\mathbf{a}_2)$ ssi $\mathbf{a}_1 = \mathbf{a}_2$
- par minimisation
 - minimiser \mathbf{a}_1 et \mathbf{a}_2
 - puis tester $\mathbf{a}_1 = \mathbf{a}_2$

Temps (avec algorithme de Hopcroft) : $O(\text{card } A \cdot n \cdot \log n)$ (ou sinon avec l'algorithme de Moore)

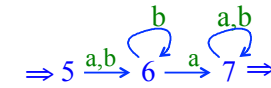
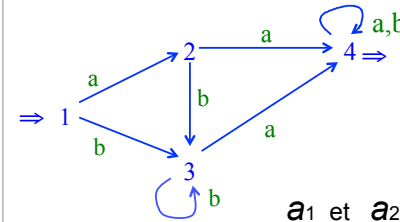
- test direct
utilise la représentation d'ensembles
avec opérations UNION / CLASSE

Temps : $O(\text{card } A \cdot n \cdot \alpha(n))$ [$\alpha = \text{"logstar"}$]

509

Exemple

UMLV ©



\mathbf{a}_1 et \mathbf{a}_2 équivalents \leftrightarrow i_1 et i_2 équivalents
(dans union disjointe)

Construction de l'équivalence :

1	2	3	4	5	6	7
1 5	2	3	4	6	7	
1 5	2 6	3	4	7		
1 5	2 3 6	4	7			
1 5	2 3 6	4	7			

\mathbf{a}_1 et \mathbf{a}_2 équivalents

510

Algorithme

UMLV ©

procédure Equiv(q_1, q_2);

début

si un seul de q_1 et q_2 terminal **alors**

« \mathbf{a}_1 et \mathbf{a}_2 non équivalents »

sinon {

$C_1 \leftarrow \text{classe}(q_1)$; $C_2 \leftarrow \text{classe}(q_2)$;

si $C_1 \neq C_2$ **alors** {
remplacer C_1 et C_2 par $C_1 \cup C_2$;

pour chaque $a \in A$ **faire**

Equiv($\delta_1(q_1, a), \delta_2(q_2, a)$);

}

Fin

Appel initial : Equiv(i_1, i_2)

Temps $O(\text{card } A \cdot n \cdot \alpha(n))$ ($n = \text{card } Q_1 + \text{card } Q_2$)

avec représentation des classes par type « UNION / CLASSE »

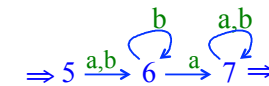
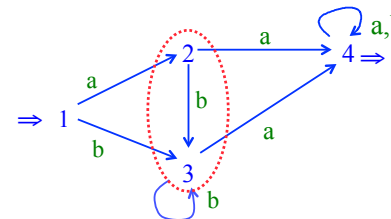
511

Minimisation par équivalence

UMLV ©

Note

Si l'un des automates est minimal, Equiv calcule la congruence syntaxique de l'autre.



\mathbf{a}_2 est minimal

Équivalence finale : 1 5 | 2 3 6 | 4 7

Congruence de \mathbf{a}_1 : 1 | 2 3 | 4

512

Flots minimaux dans les graphes

513

Réseaux de transport

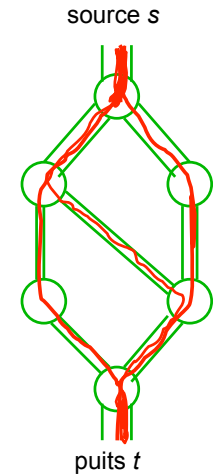
Graphe orienté valué $G = (S, A, c)$

$c(p, q)$: capacité de l'arc (p, q)

$f(p, q)$: débit ou flot de l'arc (p, q)

Exemples

- Canalisations hydrauliques
- Pipe-lines
- Voies de circulation
- Transports de marchandises,
- Réseau de communication



514

Conditions

Capacité $c : S \times S \rightarrow \mathbf{R}$ avec $c(p, q) \geq 0$
et si $(p, q) \notin A$ $c(p, q) = 0$

Flot $f : S \times S \rightarrow \mathbf{R}$

source $s \in S$, puits $t \in S$

Accessibilité

tous les sommets sont sur un chemin de s à t

Contrainte de capacité

pour tous $p, q \in S$ $f(p, q) \leq c(p, q)$

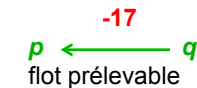
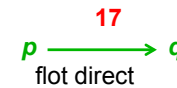


515

Conditions (suite)

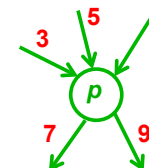
Anti-symétrie

pour tous $p, q \in S$ $f(q, p) = -f(p, q)$



Conservation du flot

pour tout $p \in S \setminus \{s, t\}$ $\sum (f(p, q) \mid q \in S) = 0$



516

Flot

UMLV ©

Valeur du flot

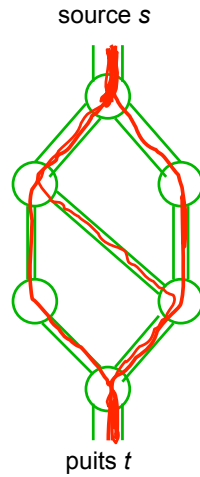
$$|f| = \sum (f(s, q) \mid q \in S)$$

ce qui part de la source

Propriété

$$|f| = \sum (f(p, t) \mid p \in S)$$

ce qui arrive au puits



517

Représentations

UMLV ©

Matrices d'adjacence

matrice des capacités
matrice des flots

Listes des successeurs

avec capacités et flots

Graphique

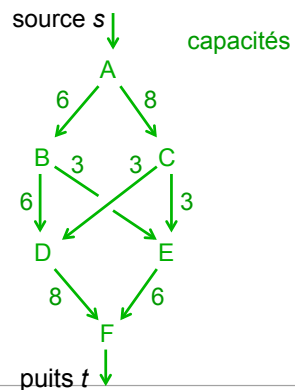


518

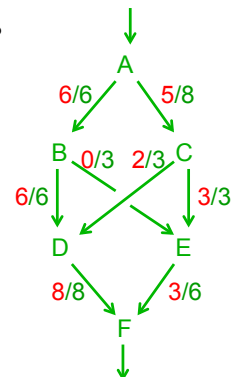
Problème

UMLV ©

Graphe orienté valué $G = (S, A, c)$
Calculer le **flot maximum**,
i.e. un flot f dont la valeur est maximale



un flot maximal ?
 $|f| = 11$



519

Méthode de Ford et Fulkerson

UMLV ©

- 1 initialiser le flot f à 0 ;
- 2 tant qu'il existe un chemin de s à t sous-utilisé faire _____ augmenter le flot f sur ce chemin ;
- 3 retour le flot f

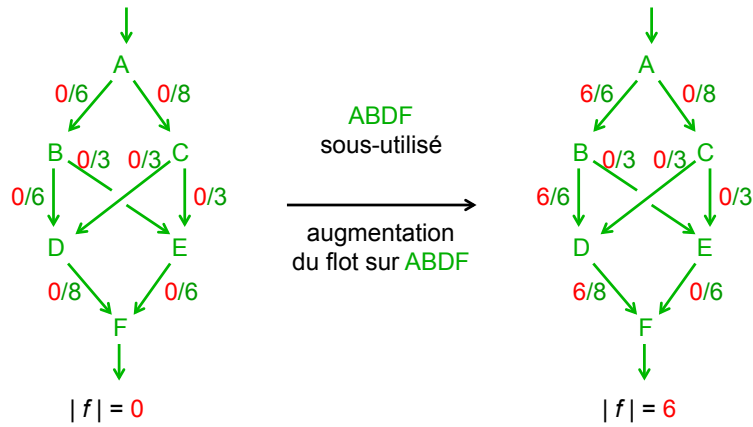
Théorème

si tout chemin $(s, \dots, p, q, \dots, p', q', \dots, t)$ possède
un arc avant plein, $f(p, q) = c(p, q)$,
ou un arc arrière vide, $f(q', p') = 0$,
alors le flot est maximum

520

Exemple

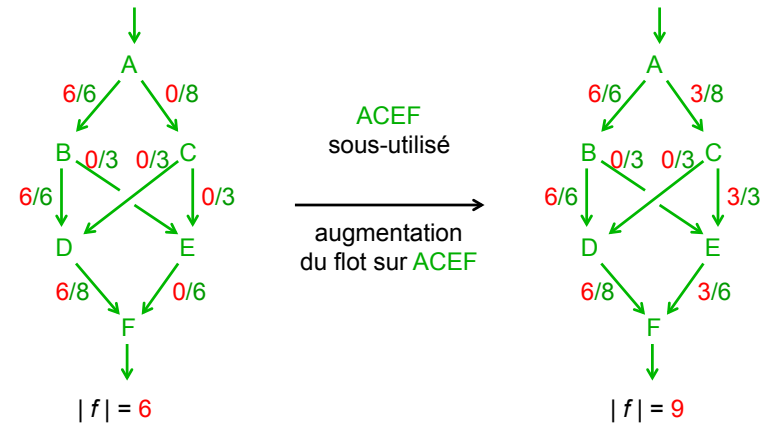
UMLV ©



521

Exemple (suite)

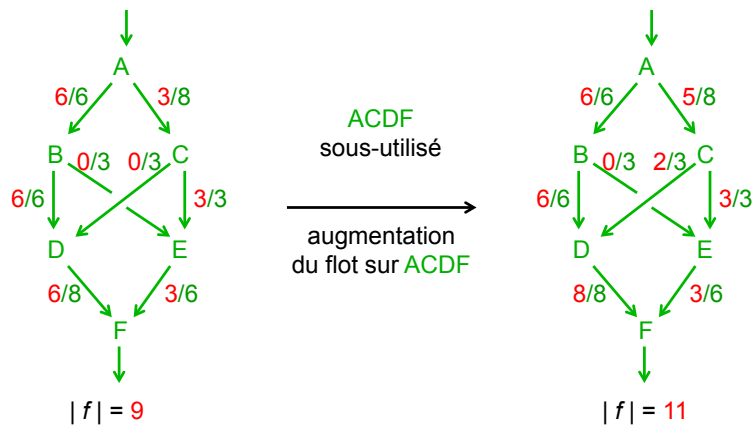
UMLV ©



522

Exemple (suite)

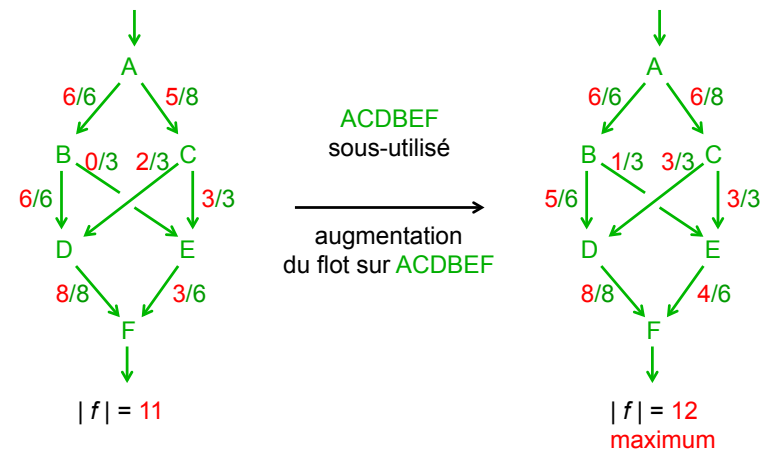
UMLV ©



523

Exemple (suite)

UMLV ©



524

Coupure

UMLV ©

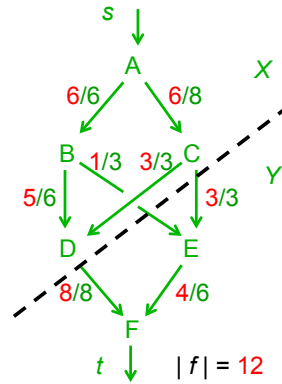
(X, Y) coupure de $G = (S, A, c)$:
 (X, Y) partition de S avec $s \in X, t \in Y$

capacité

$$c(X, Y) = \sum (c(x, y) \mid x \in X, y \in Y)$$

flot

$$f(X, Y) = \sum (f(x, y) \mid x \in X, y \in Y)$$



$$X = \{A, B, C, D\} \quad Y = \{E, F\} \quad c(X, Y) = 14 \quad f(X, Y) = 12$$

525

Propriétés (X, Y) coupure

- 1 $f(X, Y) = |f|$
- 2 $f(X, Y) \leq c(X, Y)$

Le flot maximum est borné par le minimum des capacités des coupures

- 3 f est un flot maximal ssi $|f| = c(X', Y')$ pour une coupure (X', Y')

UMLV ©

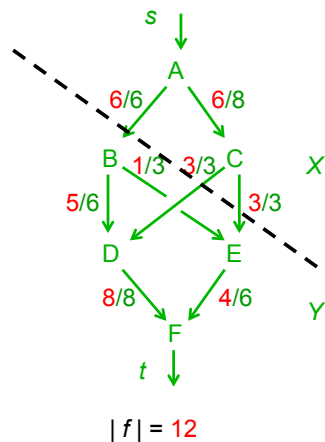
526

Coupure optimale

UMLV ©

$X' = \{A, C\}$
 $Y' = \{B, D, E, F\}$
 $c(X', Y') = 12$
 (X', Y') de capacité minimale
 $f(X', Y') = 12$
 flot maximum

Max-flow min-cut theorem



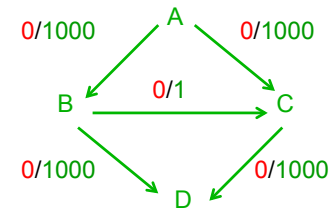
$$|f| = 12$$

527

Exemple

UMLV ©

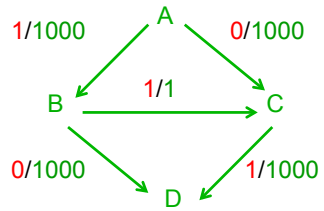
Le nombre d'itérations dépend du choix des chemins



528

Exemple

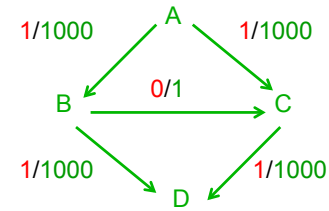
Le nombre d'itérations dépend du choix des chemins



augmentations	chemins
1	ABCD

Exemple

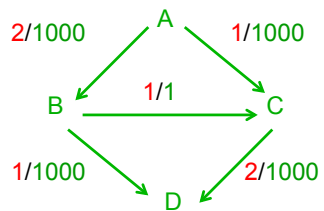
Le nombre d'itérations dépend du choix des chemins



augmentations	chemins
1	ABCD
1	ACBD

Exemple

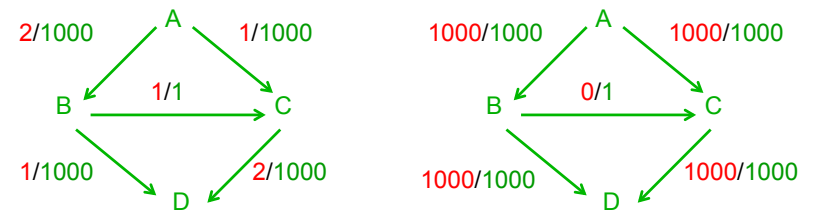
Le nombre d'itérations dépend du choix des chemins



augmentations	chemins
1	ABCD
1	ACBD
1	ABCD
	etc.

Exemple

Le nombre d'itérations dépend du choix des chemins



augmentations	chemins	augmentations	chemins
1	ABCD	1000	ABD
1	ACBD	1000	ACD
1	ABCD		flot maximum
	etc.		

Stratégie 1

UMLV ©

Pour augmenter le flot :
choisir le plus court chemin sous-utilisé (avec le parcours en largeur)

Théorème

le calcul du flot maximum obtenu avec cette stratégie nécessite au plus $\text{card}S \cdot \text{card}A$ augmentations. Le temps d'exécution total est $O(\text{card}S \cdot (\text{card}A)^2)$

Cette stratégie est connue comme *algorithme de Edmonds-Karp*

533

D'autres stratégies

UMLV ©

Push-relabel algorithm : $O((\text{card}S)^2 \cdot \text{card}A)$

Relabel-to-front algorithm : $O((\text{card}S)^3)$

534

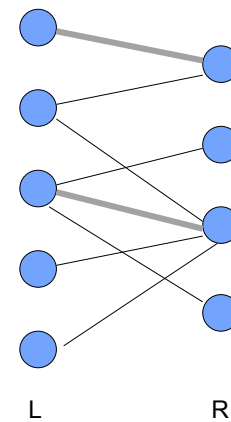
Couplage maximal dans les graphes bipartis

UMLV ©

535

Couplage maximal

UMLV ©



Graphe biparti $G = (S, A)$, $S=L \cup R$, et $\forall (p,q) \in A, p \in L$ et $q \in R$

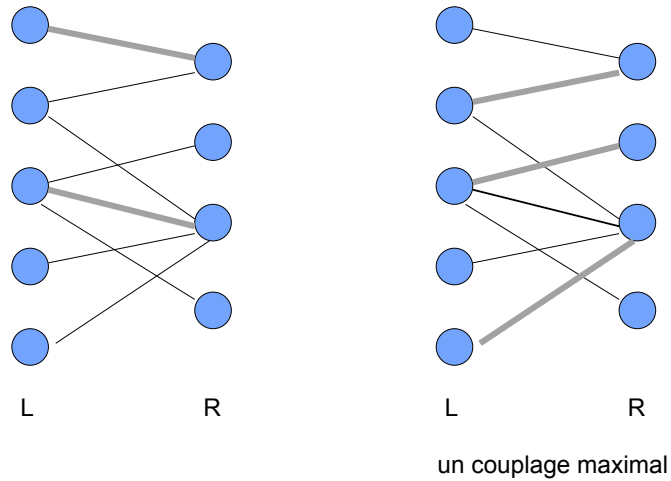
Couplage: $C \subseteq A$ tel que pour tout $p \in S$
 \exists au plus un arête de A ayant p pour extrémité

Couplage maximal: couplage contenant un nombre maximal d'arêtes (att: ce n'est pas un couplage que l'on ne peut pas étendre)

536

Couplage maximal

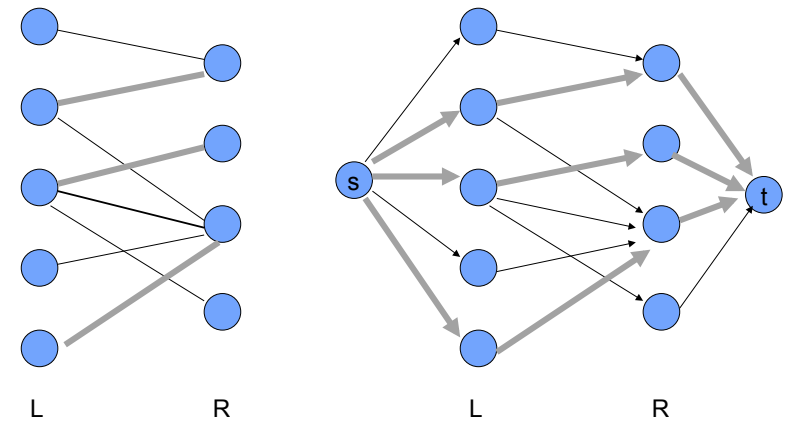
UMLV ©



537

Encodage par un flot maximal

UMLV ©



Encodage d'un graphe biparti par un graphe orienté. Couplage maximal et flot maximal correspondant. Chaque arête est de capacité 1.

538

UMLV ©

Lemma: Soit $G = (S=L \cup R, A)$ un graphe biparti et G' le graphe orienté correspondant. Si C est un couplage de G , alors il existe un flot dans G' de valeur $|C|$. Inversement, si f un flot dans G' (de valeurs entières), alors il existe un couplage dans G de cardinalité $|f|$.

L'algorithme de Ford-Fulkerson calcul un flot maximal de valeur entière pour chaque arête. On en déduit un couplage maximal.

On peut démontrer que la complexité est bornée par $O(\text{card}S \cdot \text{card}A)$.

Des améliorations et des variantes existent : par exemple, l'algorithme de Hopcroft-Karp marche en temps $O((\text{card}S)^{1/2} \cdot \text{card}A)$

539