

# Tema 7: Redes de Flujo

Modelización Matemática  
Máster en Ingeniería de Caminos, Canales y Puertos

Ignacio Montes  
Departamento de Estadística e I.O. y D.M.

1 Descripción del problema

2 Algoritmos

Consideraremos una red orientada  $R = (V, A, p)$  cumpliendo las siguientes propiedades:

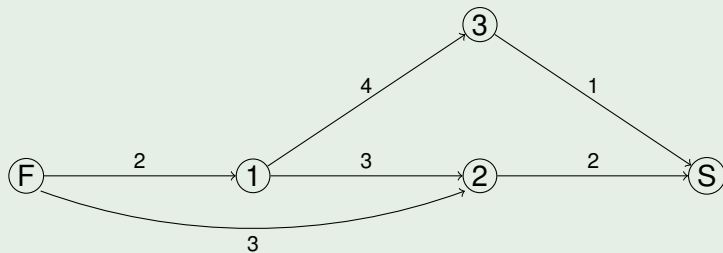
- $G = (V, A)$  es un grafo conexo.
- Los pesos son no-negativos y reciben el nombre de **capacidad** del arco.
- Hay dos vértices llamados **fuelle** ( $F$ ) y **salida** ( $S$ ) de manera que  $F$  no tiene antecesoros y  $S$  no tiene sucesores.

Partimos de una red de flujo  $R = (V, A, p)$ . El flujo de valor  $v$  es una colección de números que a cada arco  $(i, j)$  le asigna un valor  $f_{ij}$ , cumpliendo:

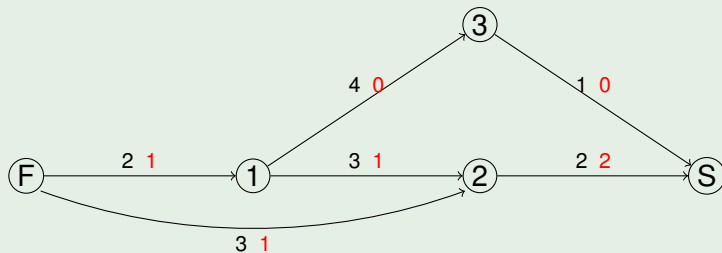
- **Acotación del flujo:**  $0 \leq f_{ij} \leq p_{ij}$ .
- **Conservación del flujo:** para todo  $i \in V$ ,

$$\sum_{j \in \Gamma(i)} f_{ij} - \sum_{k \in \Gamma^{-}(i)} f_{ki} = \begin{cases} v, & \text{si } i = F. \\ -v, & \text{si } i = S. \\ 0 & \text{en otro caso.} \end{cases}$$

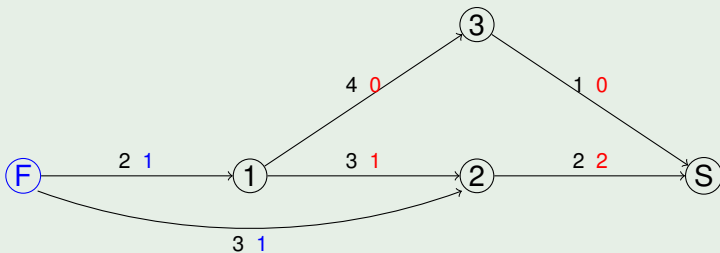
## Ejemplo (Red de flujo)



## Ejemplo (Red de flujo de valor $v = 2$ )



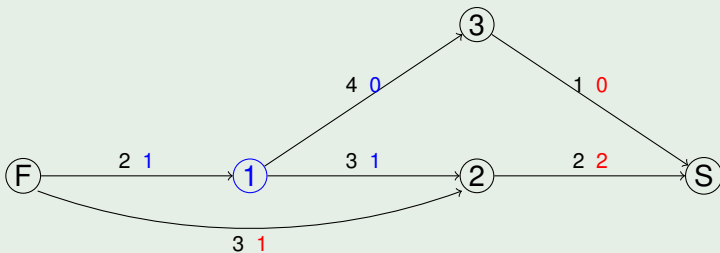
## Ejemplo (Red de flujo de valor $v = 2$ )



$$\Gamma(F) = \{1, 2\}, \Gamma^-(F) = \emptyset$$

$$\sum_{j \in \Gamma(F)} f_{Fj} - \sum_{k \in \Gamma^-(F)} f_{kF} = 1 + 1 = 2$$

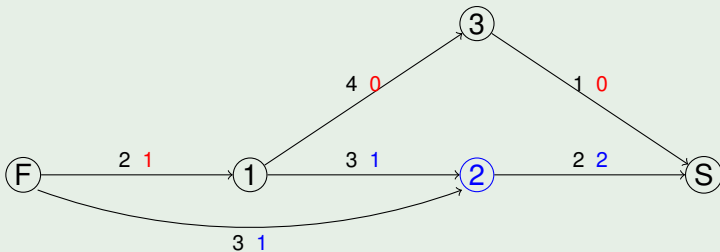
## Ejemplo (Red de flujo de valor $v = 2$ )



$$\Gamma(1) = \{2, 3\}, \Gamma^-(1) = \{F\}$$

$$\sum_{j \in \Gamma(1)} f_{1j} - \sum_{k \in \Gamma^-(1)} f_{k1} = 1 + 0 - 1 = 0$$

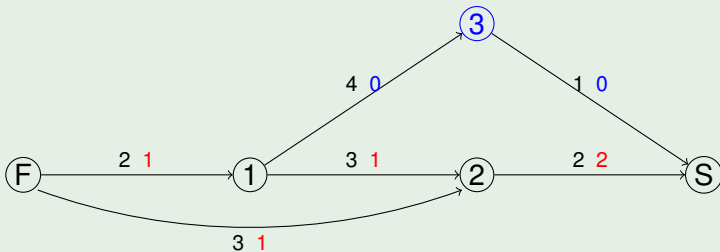
## Ejemplo (Red de flujo de valor $v = 2$ )



$$\Gamma(2) = \{S\}, \Gamma^-(2) = \{F, 1\}$$

$$\sum_{j \in \Gamma(2)} f_{2j} - \sum_{k \in \Gamma^-(2)} f_{k2} = 2 - 1 - 1 = 0$$

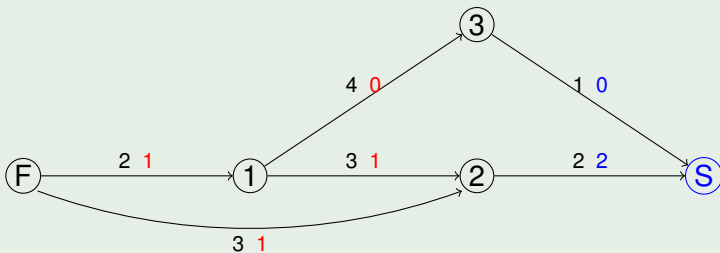
## Ejemplo (Red de flujo de valor $v = 2$ )



$$\Gamma(3) = \{S\}, \Gamma^-(3) = \{1\}$$

$$\sum_{j \in \Gamma(3)} f_{3j} - \sum_{k \in \Gamma^-(3)} f_{k3} = 0 - 0 = 0$$

## Ejemplo (Red de flujo de valor $v = 2$ )



$$\Gamma(S) = \emptyset, \Gamma^-(S) = \{2, 3\}$$

$$\sum_{j \in \Gamma(S)} f_{sj} - \sum_{k \in \Gamma^-(S)} f_{ks} = -2 - 0 = -2$$

Dada una red de flujo  $R = (V, A, p)$  y una partición  $B, B^c$  de  $V$ , el **corte** de la red de flujo se denota por  $(B, B^c)$  y es el conjunto de arcos de  $A$  con origen en  $B$  y final  $B^c$ .

- La **capacidad** de corte  $(B, B^c)$  es el valor:

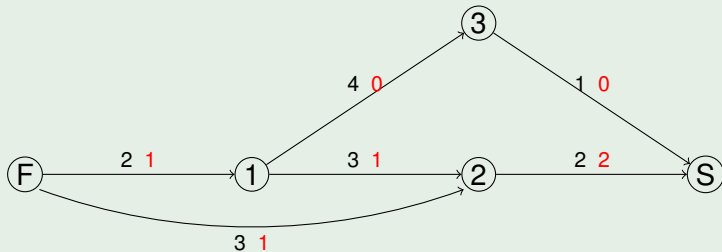
$$c(B, B^c) = \sum_{ij \in (B, B^c)} p_{ij}.$$

- El **flujo neto** del corte  $(B, B^c)$  es el valor:

$$f(B, B^c) = \sum_{i \in B, j \in B^c \cap \Gamma_i} f_{ij} - \sum_{i \in B, k \in B^c \cap \Gamma_i^-} f_{ki}.$$

# Ejemplo de corte de un flujo

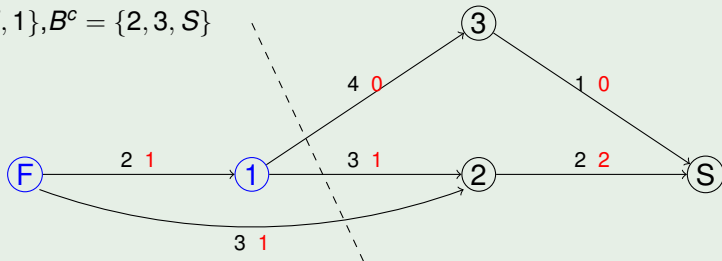
## Ejemplo (Corte )



# Ejemplo de corte de un flujo

Ejemplo (Corte  $B = \{F, 1\}, B^c = \{2, 3, S\}$ )

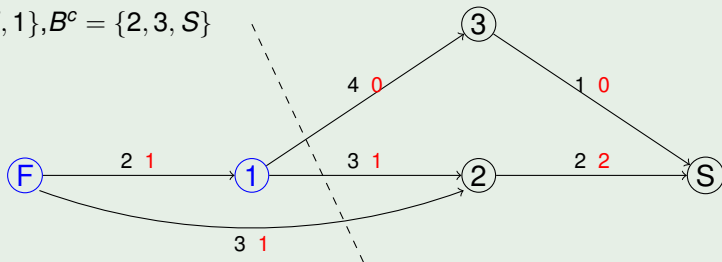
$B = \{F, 1\}, B^c = \{2, 3, S\}$



# Ejemplo de corte de un flujo

Ejemplo (Corte  $B = \{F, 1\}, B^c = \{2, 3, S\}$ )

$B = \{F, 1\}, B^c = \{2, 3, S\}$



**Capacidad:**  $c(B, B^c) = 4 + 3 + 3 = 10$ .

**Flujo neto:**  $f(B, B^c) = 0 + 1 + 1 = 2$ .

## Teorema

*Sea  $R = (V, A, p)$  una red de flujo,  $f$  un flujo y  $(B, B^c)$  un corte. Entonces:*

- *El valor del flujo está acotado por la capacidad del corte.*
- *El valor del flujo coincide con el flujo neto.*

# Objetivo: maximizar el flujo

Sea  $R = (V, A, p)$  una red de flujo.

- **Flujo máximo:** es un flujo  $f^*$  de manera que cualquier otro flujo  $f$  satisface  $v_f \leq v_{f^*}$ .
- **Objetivo:** encontrar el flujo máximo en  $R = (V, A, p)$ .

# Objetivo: maximizar el flujo

Sea  $R = (V, A, p)$  una red de flujo.

- **Flujo máximo:** es un flujo  $f^*$  de manera que cualquier otro flujo  $f$  satisface  $v_f \leq v_{f^*}$ .
- **Objetivo:** encontrar el flujo máximo en  $R = (V, A, p)$ .
- **Corte mínimo:** es un corte  $(B, B^c)$  que para otro corte  $(C, C^c)$  cumple  $c(B, B^c) \leq c(C, C^c)$ .
- Se cumple que el valor del flujo máximo es igual a la capacidad del corte mínimo.

**Ford-Fulkerson** Algoritmo clásico de etiquetado. Se trata de buscar caminos de flujo aumentable.

**Boykov-Kolmogorov** Algoritmo novedoso (2004) que mejora algoritmos previos tanto para el problema del flujo máximo como para el flujo a coste mínimo. Es el algoritmo utilizado por Mat-Lab.

# Ford-Fulkerson (I)

- 1 Considera un flujo  $f$  en la red  $R = (V, A, p)$  (por ejemplo,  $f=0$ ).
- 2 Denotar por  $A_1$  el conjunto de arcos no saturados ( $p_{ij} > f_{ij}$ ) y por  $A_2$  el conjuntos de arcos con flujo positivo ( $f_{ij} > 0$ ). Etiquetado:
  - 1 Se le asigna la etiqueta + al vértice  $F$ .
  - 2 Si  $i$  está etiquetado,  $j$  no está etiquetado y  $(i, j) \in A_1$ , entonces etiqueto  $i$  y  $(i, j)$ .  $(i, j)$  es un *arco de salida*.
  - 3 Si  $j$  no está etiquetado,  $i$  está etiquetado y  $(j, i) \in A_2$ , entonces etiquetar  $j$  y  $(j, i)$ .  $(j, i)$  es un *arco de entrada*.
  - 4 Continuar el etiquetado hasta llegar a  $S$  o hasta que no sea posible etiquetar más arcos/vértices.

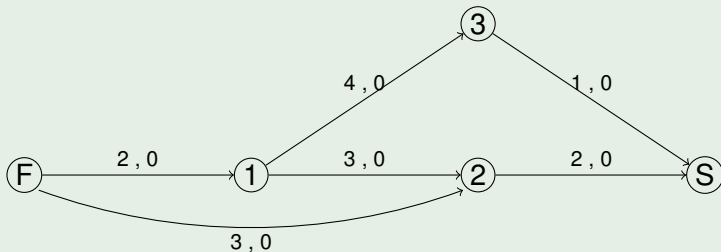
- ③ Si  $S$  no se ha etiquetado, el flujo es máximo. Si  $S$  se ha etiquetado, buscar una **cadena** formada por arcos etiquetados desde  $F$  hasta  $S$ . Definir:

$$\begin{aligned}k_1 &= \min\{f_{ij} \mid (i, j) \in C \cap A_2\}, \\k_2 &= \min\{p_{ij} - f_{ij} \mid (i, j) \in A_1 \cap C\}, \\k &= \min\{k_1, k_2\}.\end{aligned}$$

Aumentar en  $k$  unidades el flujo de los arcos de  $C \cap A_1$  y disminuir en  $k$  unidades el flujo de los arcos de  $C \cap A_2$ . Volver al paso 2.

# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 1: consideramos el flujo nulo)



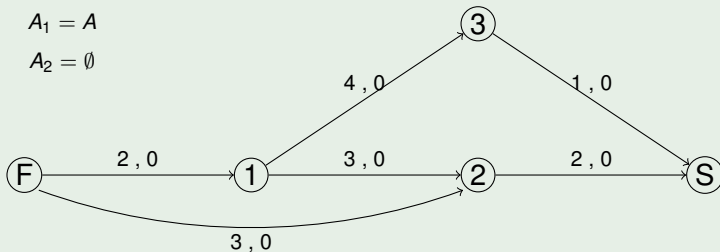
Valor del flujo:  $v = 0$

# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 2: definir $A_1, A_2$ )

$$A_1 = A$$

$$A_2 = \emptyset$$

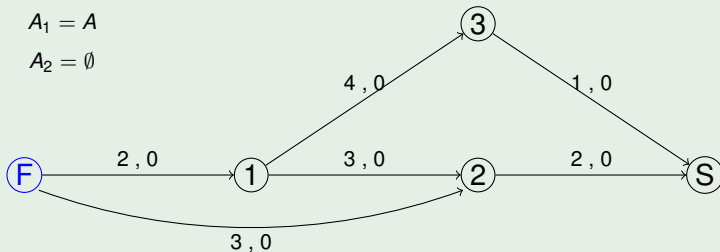


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 2.1: etiquetar el vértice $F$ )

$$A_1 = A$$

$$A_2 = \emptyset$$

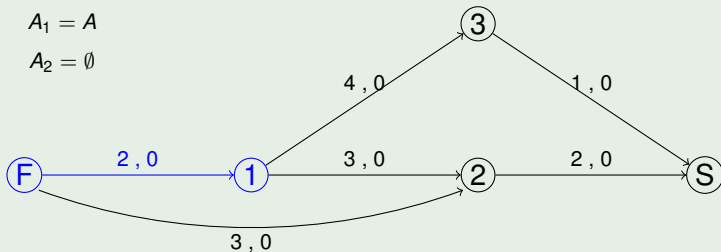


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Pasos 2.2, 2.3, 2.4)

$$A_1 = A$$

$$A_2 = \emptyset$$

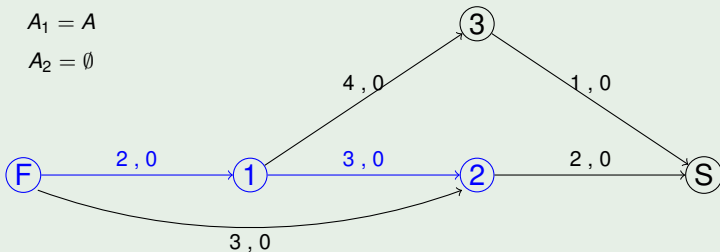


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Pasos 2.2, 2.3, 2.4)

$$A_1 = A$$

$$A_2 = \emptyset$$

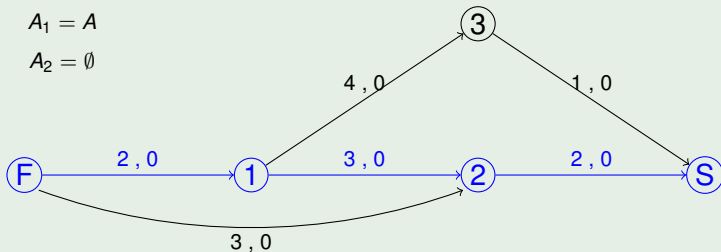


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Pasos 2.2, 2.3, 2.4)

$$A_1 = A$$

$$A_2 = \emptyset$$

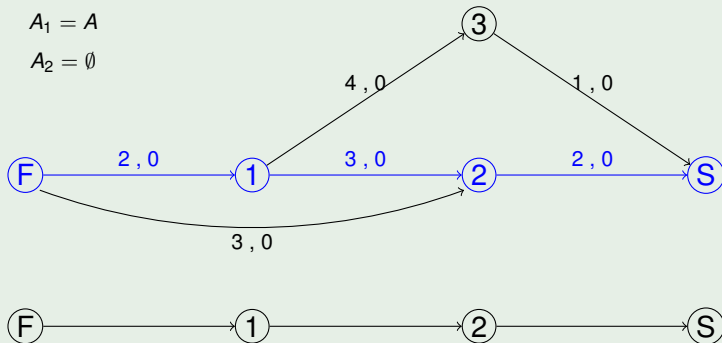


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 3: cadena de flujo aumentable)

$$A_1 = A$$

$$A_2 = \emptyset$$

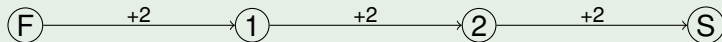
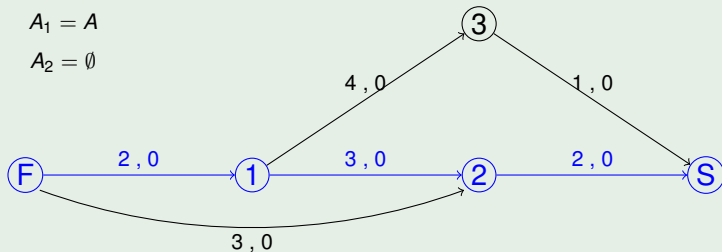


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 3: cadena de flujo aumentable)

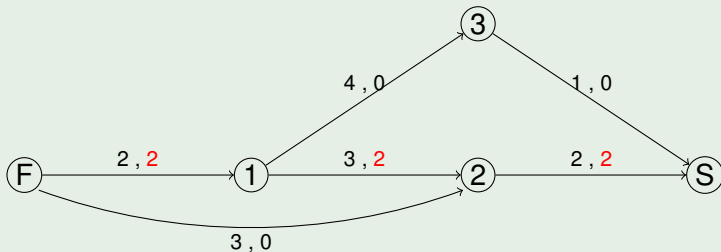
$$A_1 = A$$

$$A_2 = \emptyset$$



# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 1: actualización del flujo)



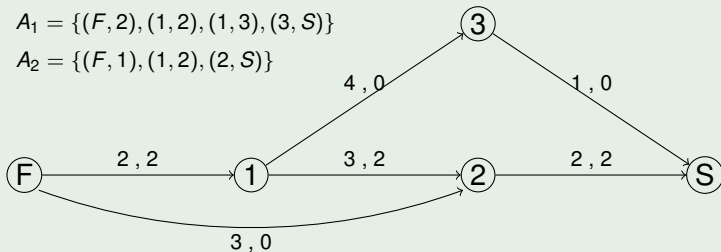
Valor del flujo:  $v = 2$

# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 2: definir $A_1, A_2$ )

$$A_1 = \{(F, 2), (1, 2), (1, 3), (3, S)\}$$

$$A_2 = \{(F, 1), (1, 2), (2, S)\}$$

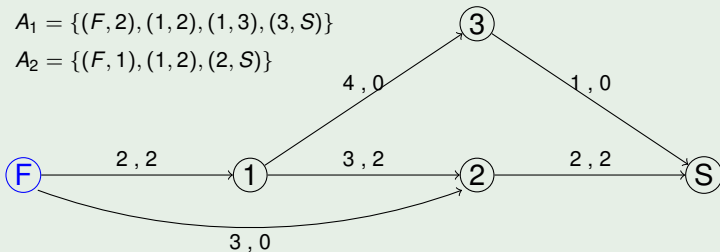


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 2.1: etiquetar el vértice $F$ )

$$A_1 = \{(F, 2), (1, 2), (1, 3), (3, S)\}$$

$$A_2 = \{(F, 1), (1, 2), (2, S)\}$$

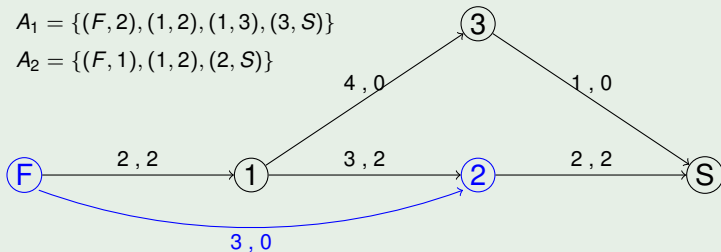


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Pasos 2.2, 2.3, 2.4)

$$A_1 = \{(F, 2), (1, 2), (1, 3), (3, S)\}$$

$$A_2 = \{(F, 1), (1, 2), (2, S)\}$$

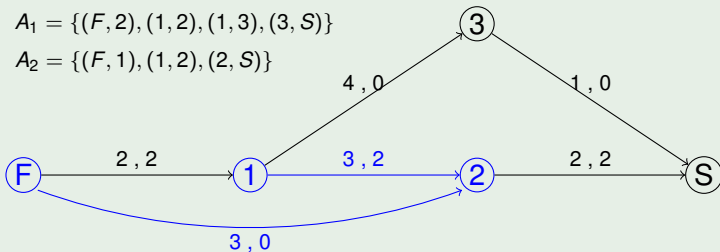


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Pasos 2.2, 2.3, 2.4)

$$A_1 = \{(F, 2), (1, 2), (1, 3), (3, S)\}$$

$$A_2 = \{(F, 1), (1, 2), (2, S)\}$$

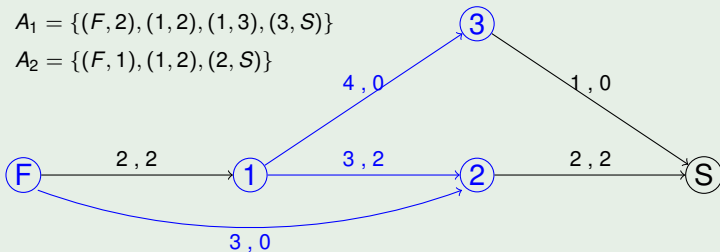


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Pasos 2.2, 2.3, 2.4)

$$A_1 = \{(F, 2), (1, 2), (1, 3), (3, S)\}$$

$$A_2 = \{(F, 1), (1, 2), (2, S)\}$$

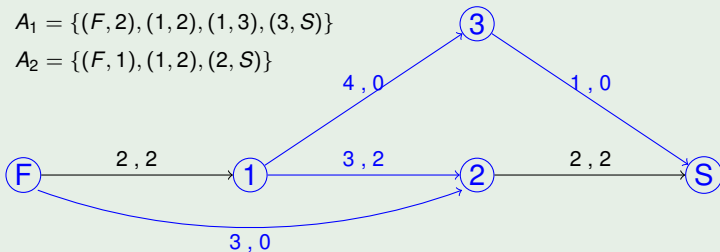


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Pasos 2.2, 2.3, 2.4)

$$A_1 = \{(F, 2), (1, 2), (1, 3), (3, S)\}$$

$$A_2 = \{(F, 1), (1, 2), (2, S)\}$$

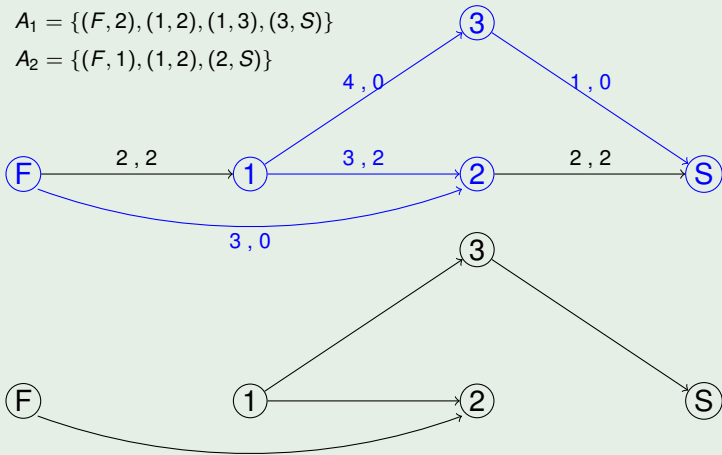


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 3: cadena de flujo aumentable)

$$A_1 = \{(F, 2), (1, 2), (1, 3), (3, S)\}$$

$$A_2 = \{(F, 1), (1, 2), (2, S)\}$$

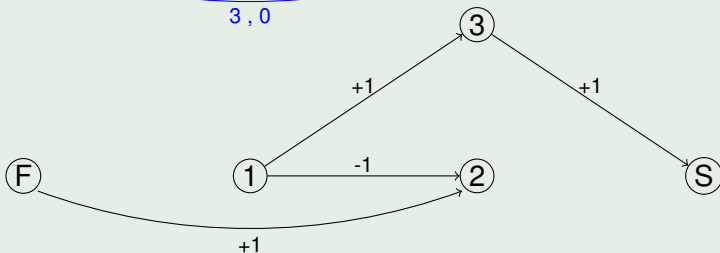
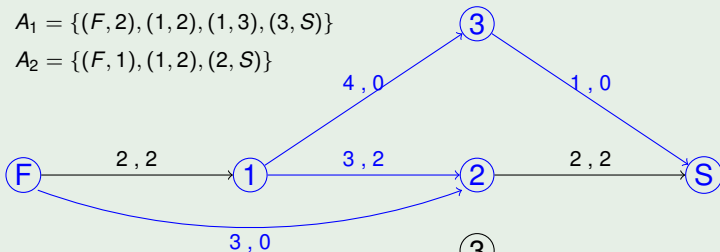


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 3: cadena de flujo aumentable)

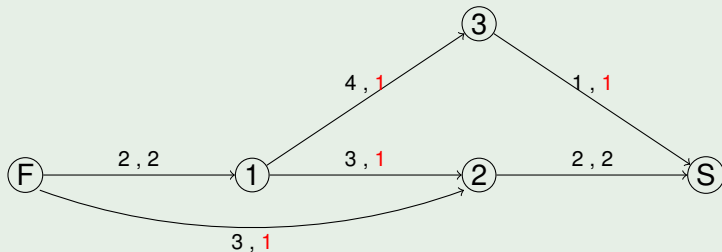
$$A_1 = \{(F, 2), (1, 2), (1, 3), (3, S)\}$$

$$A_2 = \{(F, 1), (1, 2), (2, S)\}$$



# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 1: actualización del flujo)



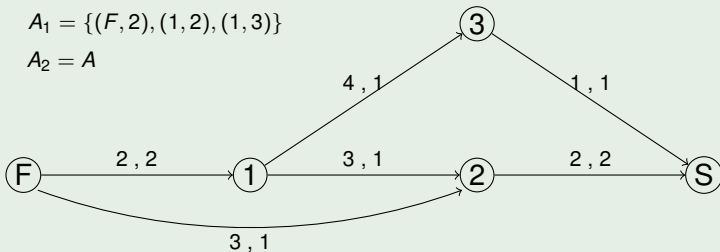
Valor del flujo:  $v = 3$

# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 2: definir $A_1, A_2$ )

$$A_1 = \{(F, 2), (1, 2), (1, 3)\}$$

$$A_2 = A$$

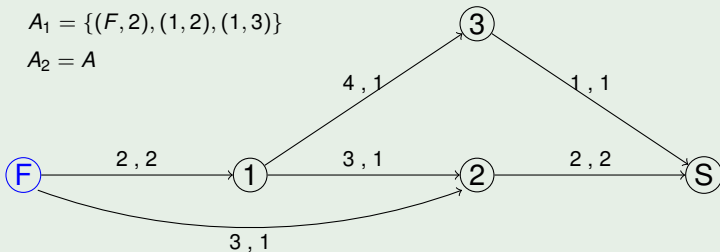


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Paso 2.1: etiquetar el vértice $F$ )

$$A_1 = \{(F, 2), (1, 2), (1, 3)\}$$

$$A_2 = A$$

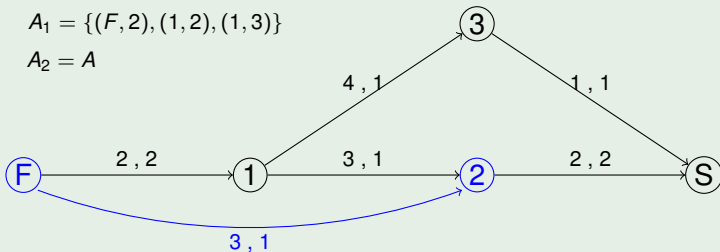


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Pasos 2.2, 2.3, 2.4)

$$A_1 = \{(F, 2), (1, 2), (1, 3)\}$$

$$A_2 = A$$

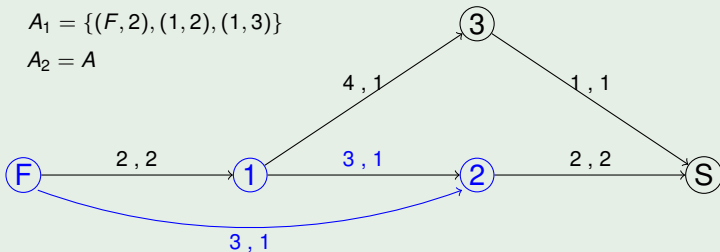


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Pasos 2.2, 2.3, 2.4)

$$A_1 = \{(F, 2), (1, 2), (1, 3)\}$$

$$A_2 = A$$

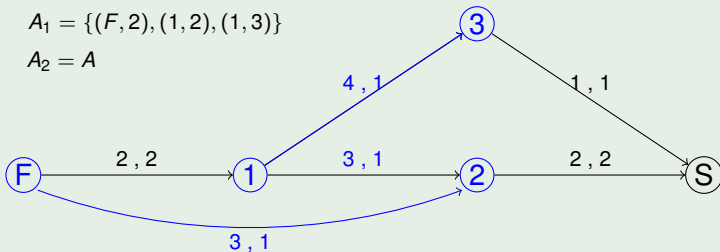


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Pasos 2.2, 2.3, 2.4)

$$A_1 = \{(F, 2), (1, 2), (1, 3)\}$$

$$A_2 = A$$

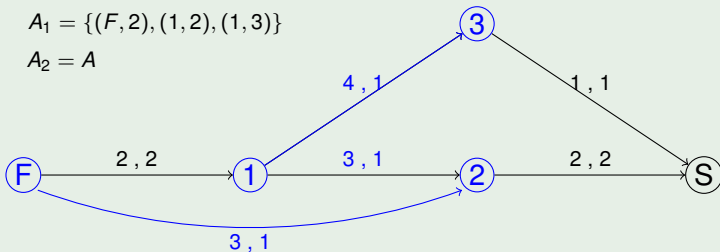


# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Pasos 2.2, 2.3, 2.4)

$$A_1 = \{(F, 2), (1, 2), (1, 3)\}$$

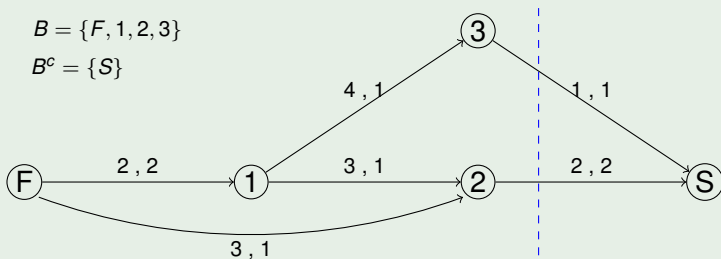
$$A_2 = A$$



No se puede etiquetar más. Flujo máximo

# Algoritmo de Ford-Fulkerson: Ejemplo

## Ejemplo (Corte mínimo)



$$c(B, B^c) = 2 + 1 = 3 = f(B, B^c)$$

# Algoritmo de Ford-Fulkerson: Ejercicio

Buscar el flujo máximo en la siguiente red de flujo:

