Alberi

IASD a.a. 2015-16

Annalisa De Bonis

Concetto di albero

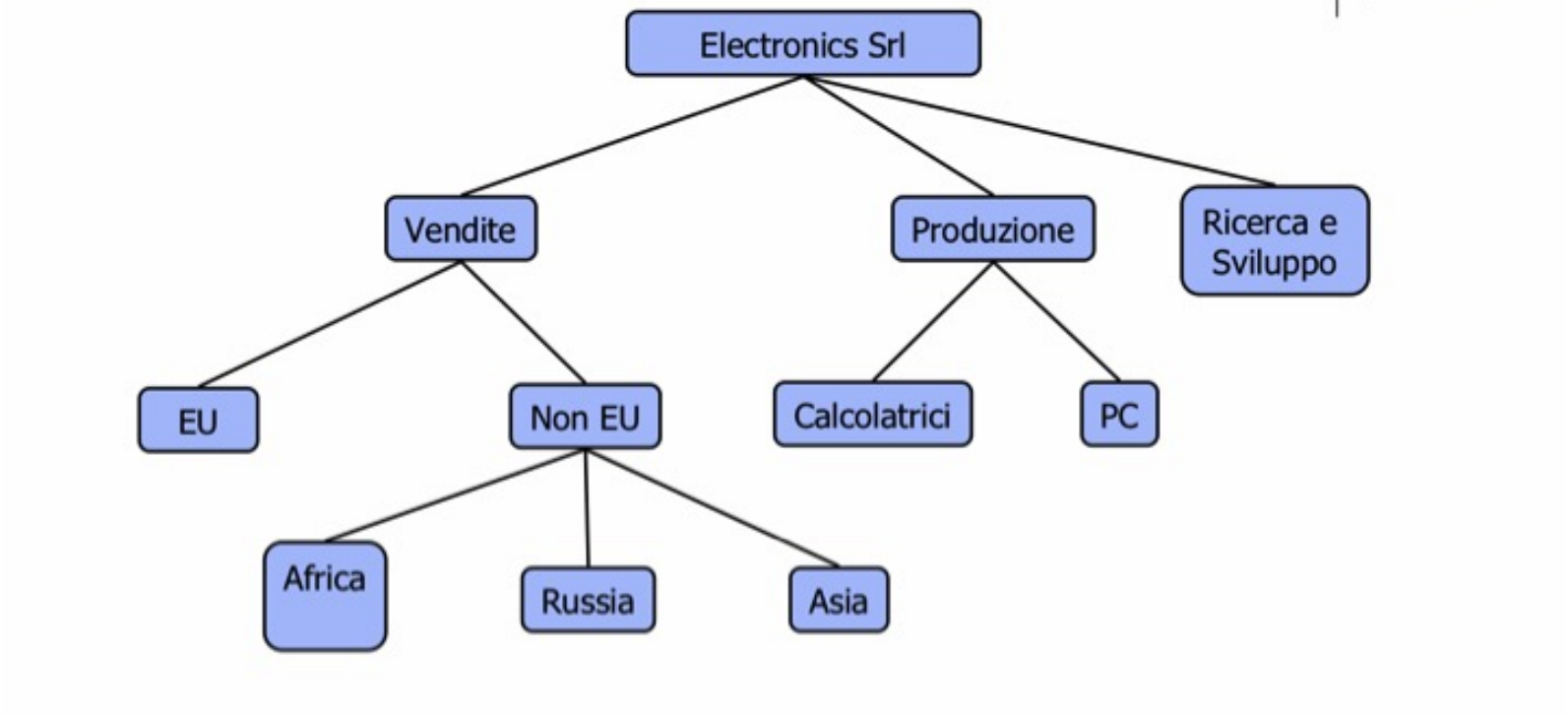
- Astrae il concetto di gerarchia
- Un albero consiste di un insieme di nodi tra cui vi è una relazione del tipo padre-figlio che soddisfa le seguenti proprietà:
  - se **T** non è vuoto allora contiene un nodo speciale chiamato **radice** che non ha padre
  - ciascun nodo diverso dalla radice ha un unico padre; i nodi aventi un certo nodo w come padre si dicono figli di w

Applicazioni

- La struttura ad albero rappresenta un modo naturale per organizzare alcuni tipi di informazioni:
  - Organigramma aziendali
  - File system
  - Siti Web
  - Ecc.

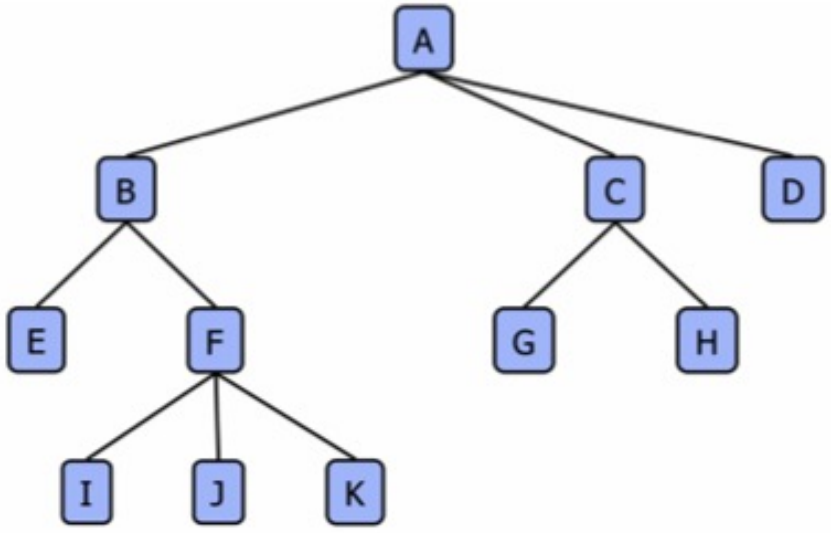
Alberi

Esempio



Alberi

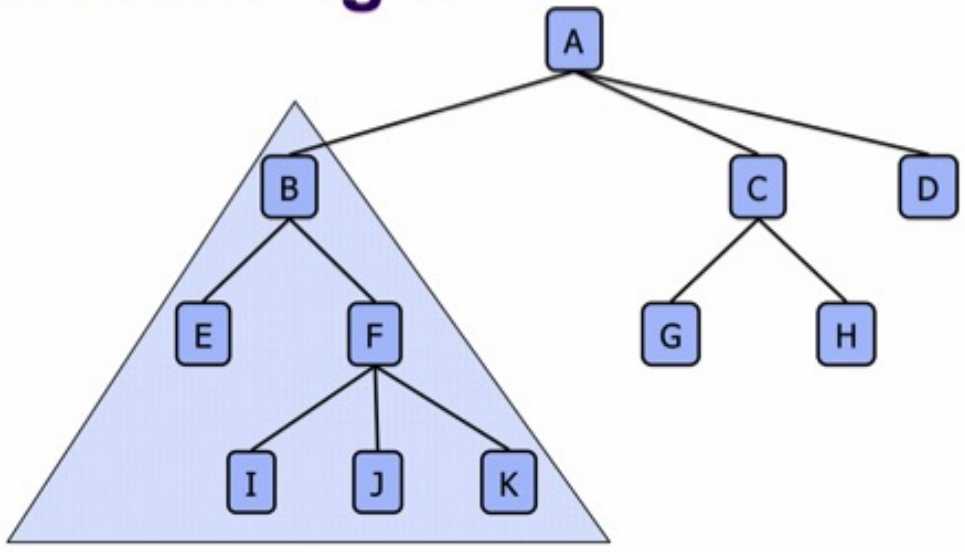
Terminologia



Radice: nodo senza genitore (A)
Nodo interno: nodo con almeno un figlio (A,B,C,F)
Foglia (nodo esterno): nodo senza figli (E, I, J, K, G, H, D)

Alberi

Terminologia

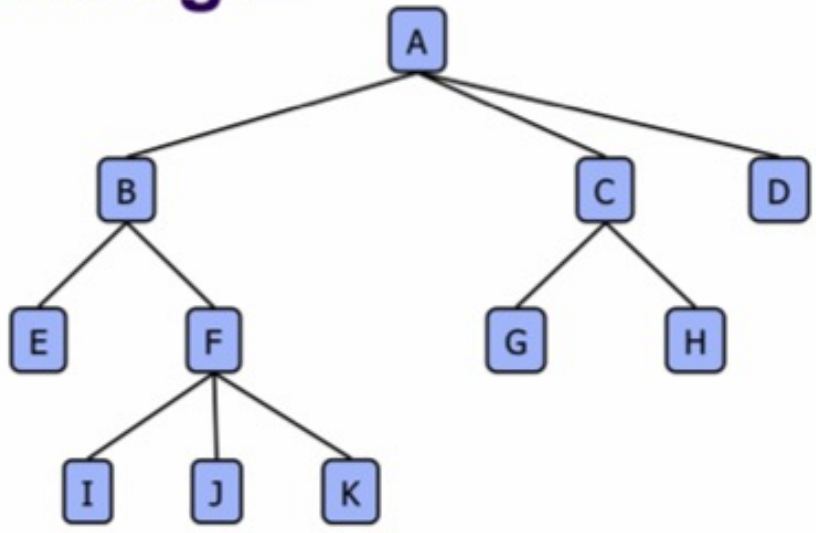


Discendente di un nodo: il nodo stesso o discendente di un figlio del nodo (discendenti di B : B,E,F,I,J,K)

Sottoalbero: albero formato da un nodo e tutti i suoi discendenti (sottoalbero con radice in B: B,E,F,I,J,K)

Alberi

Terminologia

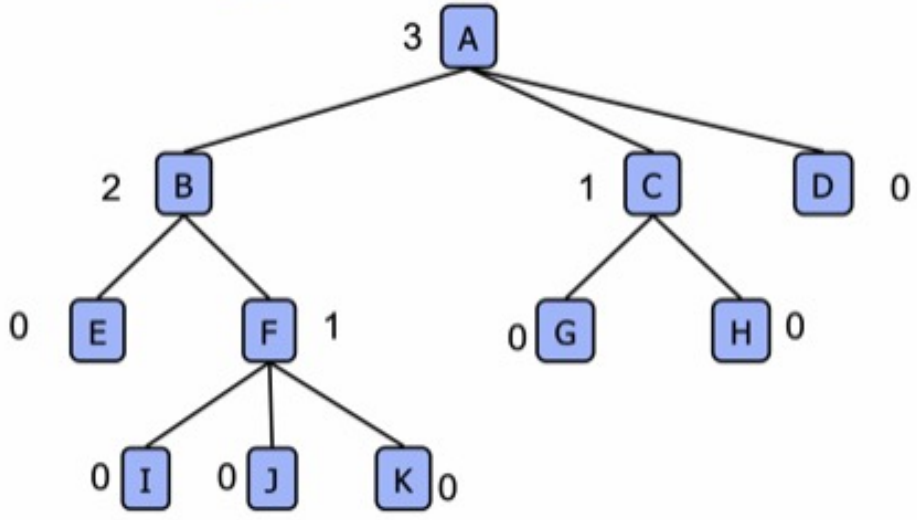


Antenato: il nodo stesso o antenato del genitore del nodo (G:G, A,C)

Profondità di un nodo: numero di antenati del nodo -1 (G: 2)

Alberi

Terminologia

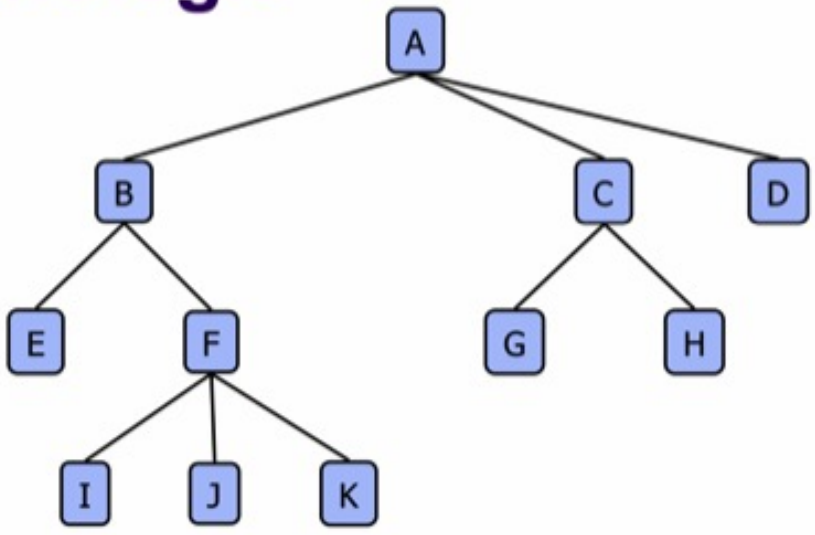


Altezza di un nodo: 0 se il nodo è una foglia;
altrimenti 1 + max altezza figli del nodo (C: 1)

Altezza di un albero non vuoto = altezza radice (3)

Alberi

Terminologia



Arco: coppia di nodi che sono in relazione padre-figlio

Percorso: sequenza di nodi in cui due nodi consecutivi sono in relazione padre-figlio (percorso da B a J: B,F,J)

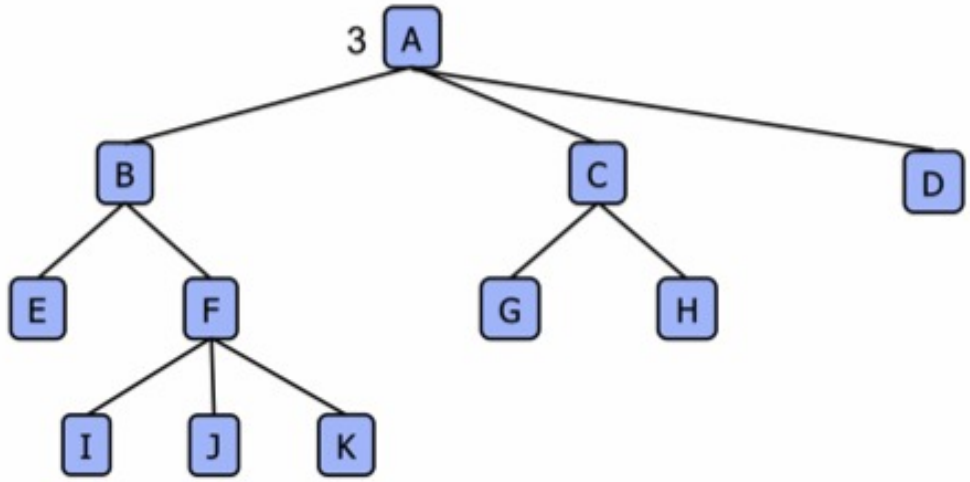
Profondità nodo = numero di archi formati dai nodi lungo il percorso dalla radice al nodo (profondità di G: 2)

Alberi

Proposizione

- Altezza di un albero non vuoto = profondità massima delle foglie

  - Giustificazione: Altezza dell'albero = numero di archi lungo il percorso dalla radice alla foglia più distante



• Si può dimostrare per induzione sul numero di nodi

Proposizione

- Definiamo
  - c_v = numero di figli di v
  - T_w = sottoalbero con radice w
  - $|T_w|$ = numeri di nodi in T_w
- Si ha che $\sum_{v \in T_w} c_v = |T_w| - 1$

  Dim. Ad eccezione di w, ciascun nodo in T_w è figlio di un unico nodo di T_w e quindi dà un contributo pari ad 1 alla sommatoria.

Alberi cardinali e ordinali

Alberi cardinali

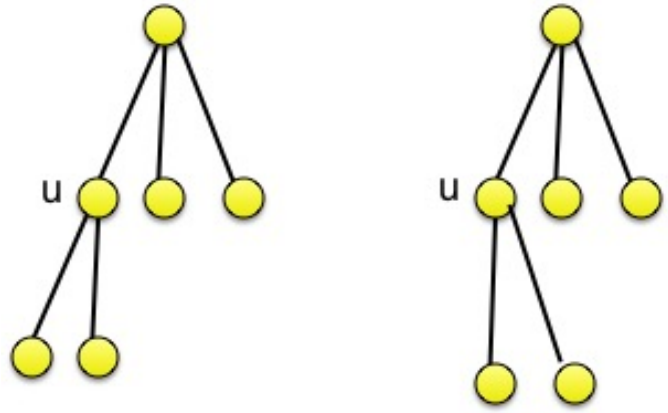
- Ogni nodo ha k riferimenti ai figli, i quali sono enumerati da 0 a $k-1$
- I riferimenti ai figli di un nodo sono memorizzati in un array di dimensione k
- Elemento i-esimo dell'array: riferimento (eventualmente uguale a `null`) all'i-esimo figlio
- Accesso diretto a ciascun figlio

Alberi ordinali

- Ogni nodo memorizza soltanto la lista ordinata dei riferimenti *non vuoti* ai suoi figli.
- Accesso sequenziale a ciascun figlio : tempo $O(i)$ per accedere al figlio i

Alberi cardinali e ordinali

- Gli alberi cardinali e ordinali sono due strutture di dati differenti



- I due alberi in alto sono uguali se considerati come alberi ordinali e diversi se considerati come alberi cardinali perché nel primo albero il nodo u non ha il figlio numero 2 mentre nel secondo albero il nodo u non ha il figlio numero 0.

Profondità di un nodo:versione ricorsiva

```
Profondita( u ):
  IF (u.padre == null) {
    RETURN 0;
  }
  pp= Profondita(u.padre)
  RETURN pp+1;
```

- **Tempo**: il tempo di esecuzione, escluso il tempo per la chiamata ricorsiva, è $O(1)$. Indichiamo con $T(x)$ il costo della chiamata $Profondita(x)$.
  - $T(x) = O(1) + T(padre(x))$
  - $T(padre(x)) = O(1) + T(padre(padre(x))$
  - $T(padre(padre(x)) = O(1) + T(padre(padre(padre(x)))$
  - ...
  - $T(radice) = O(1)$
- $T(x) = O(1) \times$ numero chiamate ricorsive

- Sia d la profodità di x. In totale, l'invocazione di Profondita(x) innesca una catena di d chiamate ricorsive $\implies$ tempo totale $T(x) = O(d)$.

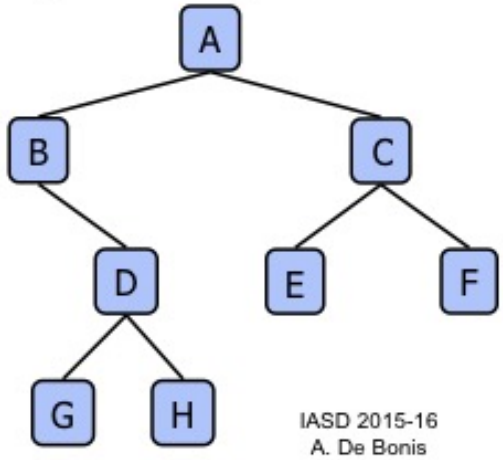
Profondità di un nodo: versione iterativa

```
Profondita( u ):
  p = 0;
  WHILE (u.padre != null) {
    p = p + 1;
    u = u.padre;
  }
```

Alberi binari

Albero binario

- Albero binario :
  - Ogni nodo interno ha al piu` due figli (esattamente due se l'albero e` un albero binario ***proprio***)
  - I figli di un nodo sono **una coppia ordinata**
- I figli di un nodo interno sono chiamati figlio destro e figlio sinistro
- Definizione alternativa (ricorsiva):
  - Un albero vuoto, oppure
  - Un albero costituito da una radice che ha una coppia ordinata di sottoalberi ognuno dei quali è un albero binario

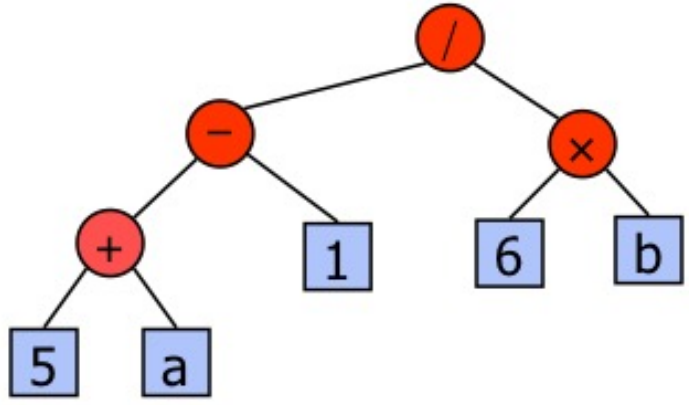


Alberi binari

- Gli alberi binari sono un caso particolare degli alberi cardinali in cui $k = 2$
- Per ogni nodo interno, il figlio 0 corrisponde al figlio sinistro e il figlio 1 corrisponde al figlio destro.

Applicazioni degli alberi binari:
albero sintattico per le espressioni

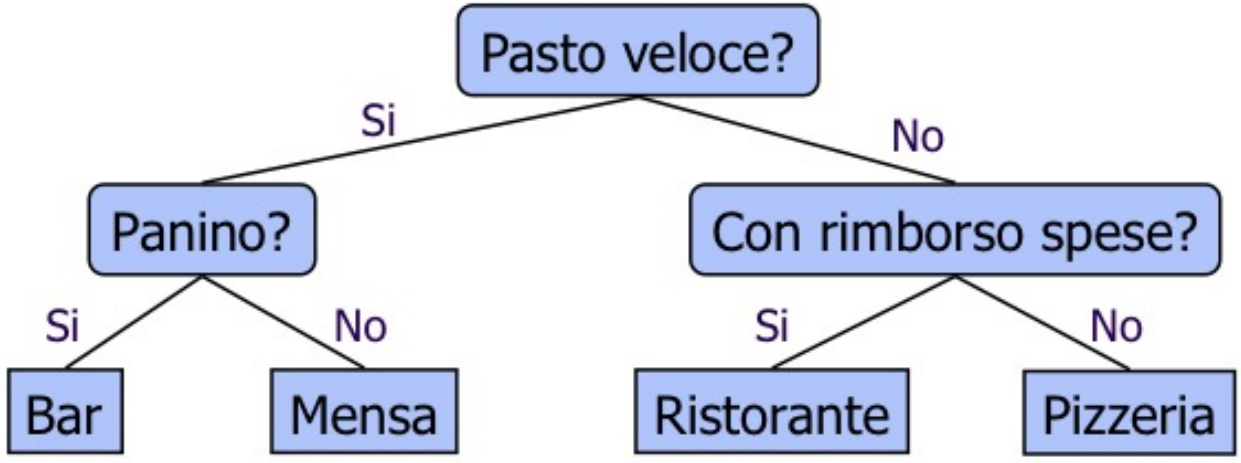
- Albero binario associato ad una espressione:
  - Nodi interni: operatori
  - Nodi esterni: operandi
- Esempio: $((5 + a) - 1) / (6 \times b)$



Alberi binari

Applicazioni degli alberi binari:
Albero di decisione binario

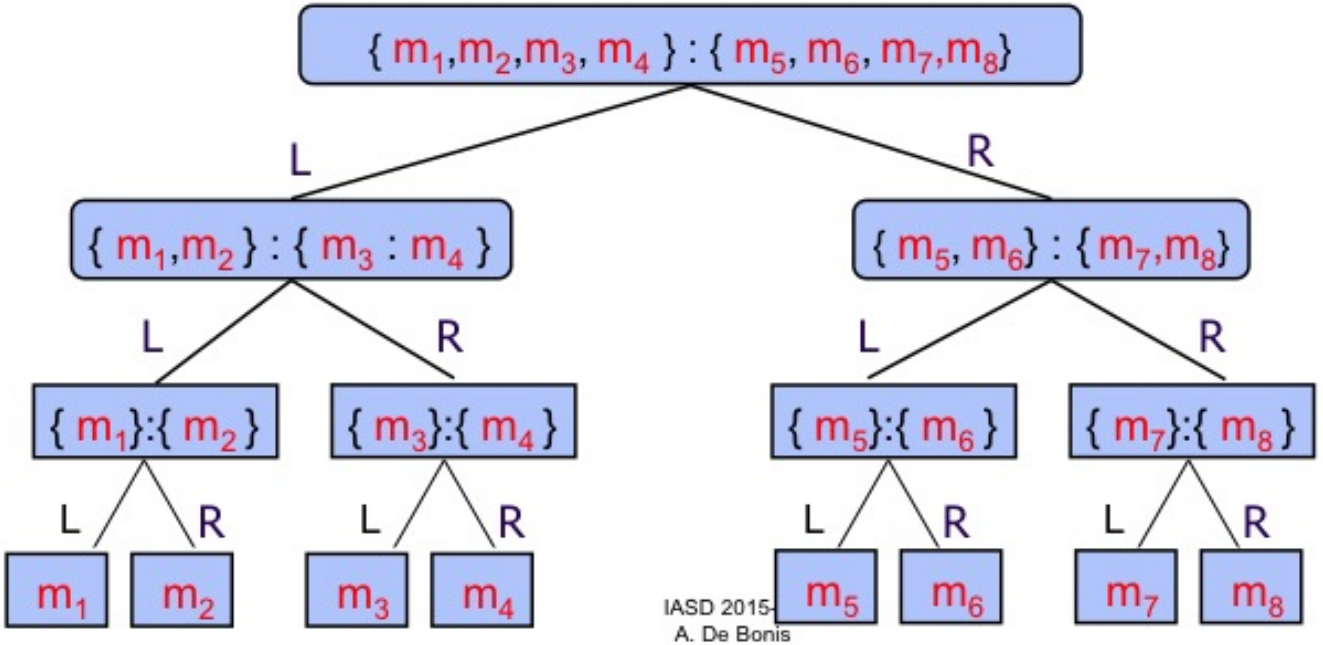
- Albero binario associato ad un processo di decisione:
  - Nodi interni: domande con 2 possibili risposte: SI/NO
  - Nodi esterni: decisioni
- Esempio: schema di decisione per un pasto



Alberi binari

Albero di decisione binario

- **Esempio:** schema di decisione per trovare una moneta falsa (più pesante) tra 8 monete apparentemente identiche $\{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8\}$
- usiamo una bilancia a due piatti
  - Possibile risposte: L = bilancia pende a sinistra
    R = bilancia pende a destra



Alberi binari

- Un nodo ha i campi
  - dato
  - padre
  - sx (figlio sinistro)
  - dx (figlio destro)