SED temporizzati e non-temporizzati

- **SED non-temporizzati**: l'ingresso è specificato come una sequenza di eventi $\{e_1, e_2, e_3, \dots\}$.

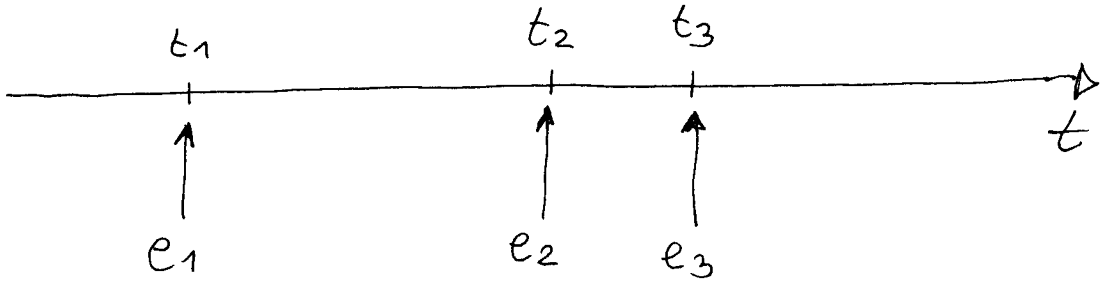
$$\{e_1, e_2, e_3, \dots\} \longrightarrow \boxed{\text{SED}} \longrightarrow \{x_0, x_1, x_2, x_3, \dots\}$$

Si può determinare la traiettoria, ma non il moto, del sistema.

Nel caso dei SED non temporizzati è nota solo la successione cronologica degli eventi, non gli istanti di tempo in cui gli eventi si verificano.

Per SED non-temporizzati, possiamo studiare la logica di funzionamento (per questo sono anche detti SED LOGICI)

- **SED temporizzati**: l'ingresso è specificato come una sequenza di eventi con i rispettivi tempi di occorrenza $\{(e_1, t_1), (e_2, t_2), \dots\}$



$\Rightarrow$ si può determinare il moto del sistema.

MODELLI DI SISTEMI AD EVENTI DISCRETI non-temporizzati

↘ Molti modelli ad eventi discreti sono basati sul formalismo dei GRAFI.

GRAFI

Un grafo G è una coppia (V, A) dove:

- $V = \{v_1, \dots, v_n\}$ è l'insieme dei vertici (nodi)
- $A = \{a_1, \dots, a_m\}$ è l'insieme degli archi

Gli archi descrivono relazioni tra vertici:

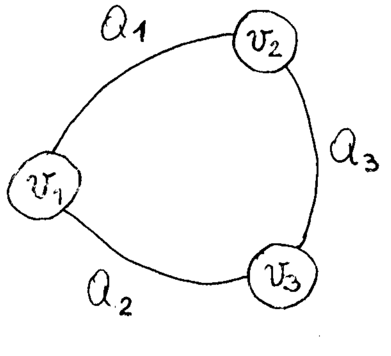
- CASO DI GRAFO NON ORIENTATO:

  $a_\ell = \{v_i; v_j\} \Rightarrow$ relazione tra v_i e v_j

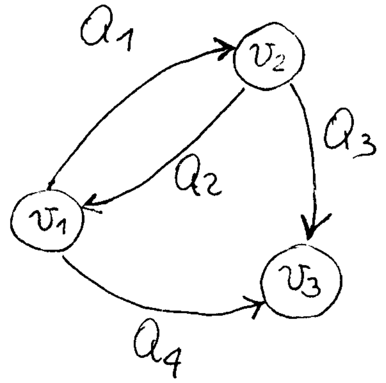
- CASO DI GRAFO ORIENTATO

  $a_\ell = (v_i; v_j) \Rightarrow$ relazione tra v_i (NODO DI PARTENZA) e v_j (NODO DI ARRIVO)

  NOTA - $A \subseteq V \times V$



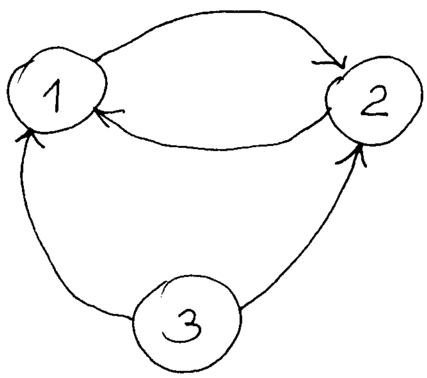
GRAFO NON ORIENTATO



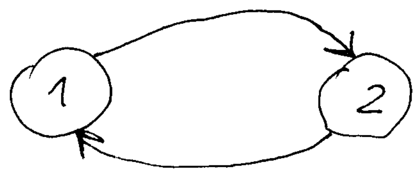
GRAFO ORIENTATO

Dati due grafi $G(V,A)$ e $H(W,B)$, H e' un <u>sottografo</u> di G se

1) $W \subseteq V$
2) $B \subseteq A$
3) $\forall\, v_i, v_j \in W$ tali che $(v_i, v_j) \in A$, allora $(v_i, v_j) \in B$.

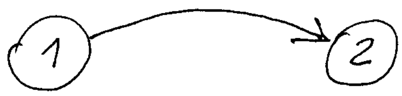


G



H e' un sottografo di G.

Se valgono solo 1) e 2), H e' un <u>grafo parziale</u> di G.



H e' un grafo parziale di G.

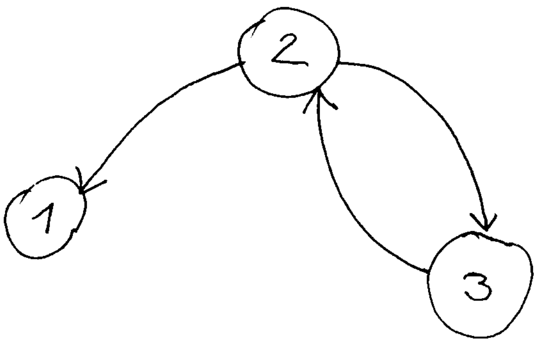
Definizione - Un cammino in un grafo G è una sequenza $\gamma = v_{i_0}\, a_{j_1}\, v_{i_1}\, a_{j_2} \dots a_{j_k}\, v_{i_k}$ dove:

- GRAFO NON ORIENTATO: $a_{j_\ell} = \{v_{i_{\ell-1}}, v_{i_\ell}\}$ $\ell = 1, \dots, k$
- GRAFO ORIENTATO: $a_{j_\ell} = (v_{i_{\ell-1}}, v_{i_\ell})$ $\ell = 1, \dots, k$.

La lunghezza del cammino γ è k (numero degli archi lungo il cammino).

Un cammino si dice ciclo se $v_{i_0} = v_{i_k}$.

Definizione - Un grafo G orientato si dice debolmente connesso se per ogni coppia $(v_i; v_j)$ esiste un cammino non orientato che collega v_i e v_j.



debolmente connesso, ma non fortemente connesso.

Definizione - Un grafo orientato si dice fortemente connesso se per ogni coppia $(v_i; v_j)$ esiste un cammino orientato che collega v_i e v_j.

<u>Definizione</u> - In un grafo non orientato, <u>due vertici v_i e v_j sono adiacenti</u> se $\{v_i; v_j\} \in A$.

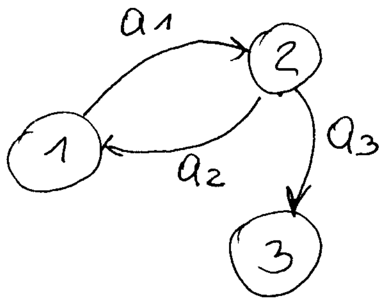
In un grafo orientato, <u>il vertice v_j è adiacente al vertice v_i</u> se $(v_i; v_j) \in A$.

RAPPRESENTAZIONI DI GRAFI

1) <u>matrici di connessione</u>

È una matrice $n \times n$ (dove n è il numero di vertici) il cui elemento di posizione $(i;j)$ è pari a 1 se $(v_i; v_j) \in A$; altrimenti è uguale a 0.

Ovviamente, per grafi non orientati la matrice di connessione è simmetrica.



$$M_c = \begin{array}{cc} & \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ & \left[\begin{array}{ccc} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{array}\right] \begin{array}{c} v_1 \\ v_2 \\ v_3 \end{array} \end{array}$$

$\Rightarrow$ occorrono n^2 posizioni di memoria per memorizzare M_c.

iii) matrici di incidenza

È una matrice $n \times m$ (dove n è il numero di vertici e m è il numero di archi) il cui elemento di posizione (i, j) è pari a:

GRAFI NON ORIENTATI

- 1 se l'arco a_j insiste sul nodo v_i
- 0 altrimenti

GRAFI ORIENTATI

- 1 se l'arco a_j parte dal nodo v_i
- −1 se l'arco a_j arriva nel nodo v_i
- 0 altrimenti

⇒ Se a_j parte e arriva nello stesso nodo v_i, allora per convenzione, si pone = 1.

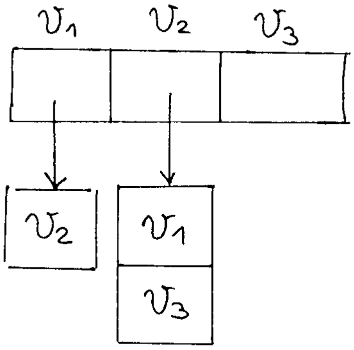
$$M_I = \begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 0 & -1 \end{bmatrix} \end{array} \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array}$$

(nell'esempio precedente)

⇒ occorrono $n \cdot m$ posizioni di memoria per memorizzare M_I.

iii) lista delle adiacenze

A ogni vertice viene associata la lista dei vertici adiacenti.



(nell'esempio precedente)

$\Rightarrow$ occorrono $n+m$ posizioni di memoria per memorizzare la lista delle adiacenze.

RELAZIONE TRA SED NON TEMPORIZZATI E LINGUAGGI

IDEA: lo spazio degli eventi è l'alfabeto di un linguaggio;
una sequenza di eventi ammissibile è una parola
di tale linguaggio.

esempio - Consideriamo una macchina che lavora un solo pezzo alla volta.
La macchina si può guastare solo durante l'esecuzione di una lavorazione.
L'insieme degli eventi è $\{\alpha, \beta, \gamma, \delta\}$ dove:

- α: inizio lavorazione
- β: termine lavorazione
- γ: guasto
- δ: riparazione

La sequenza di eventi $\alpha\beta\alpha\delta\beta$ non è ammissibile, perchè non si può verificare una riparazione se prima non si è verificato un guasto.
$\Rightarrow$ la parola $\alpha\beta\alpha\delta\beta$ non appartiene al linguaggio.

La parola $\alpha\beta\alpha\gamma\delta\alpha\beta$ appartiene invece al linguaggio.

Ricapitolando:

evento $\longleftrightarrow$ simbolo di un alfabeto

evoluzione di un SED non temporizzato $\longleftrightarrow$ parola

comportamento di un SED non temporizzato $\Leftrightarrow$ linguaggio

(cioè, l'insieme di tutte le possibili evoluzioni del sistema)

LINGUAGGI

ALFABETO: insieme finito e non vuoto di simboli, $\mathcal{E}$.

PAROLA (STRINGA): sequenza di simboli appartenenti ad $\mathcal{E}$.

LINGUAGGIO: un insieme di parole formate a partire dai simboli in $\mathcal{E}$.

NOTAZIONE: La parola vuota (sequenza di lunghezza zero) viene tipicamente indicata con ε.

Operazioni sui linguaggi

Dati un alfabeto $\mathcal{E}$, e due linguaggi $\mathcal{A}$ e $\mathcal{B}$ definiti su $\mathcal{E}$:

1. CONCATENAZIONE

$$AB = \{ w = uv : u \in A \text{ e } v \in B \}.$$

2. UNIONE E INTERSEZIONE

i) $A \cup B = \{w : w \in A \text{ oppure } w \in B\}$

ii) $A \cap B = \{w : w \in A \text{ e } w \in B\}$

3. CHIUSURA DI KLEENE

$$A^* = \{\varepsilon\} \cup A \cup A^2 \cup A^3 \dots = \bigcup_{i=0}^{\infty} A^i$$

dove:

- $A^0 = \{\varepsilon\}$
- $A^k = AA^{k-1}$ per $k = 1, 2, \dots$

Esempio

$\mathcal{L} = \{a, abb\}$ definito su $\Sigma = \{a, b\}$

$$\mathcal{L}^* = \{\varepsilon, \underbrace{a, abb}_{\mathcal{L}}, \underbrace{aa, aabb, abba, abbabb}_{\mathcal{L}^2}, \dots\}$$

LINGUAGGI REGOLARI

Definizione - Una Espressione Regolare (ER) su un alfabeto Σ e' costruita in base alle seguenti regole:

espressioni regolari atomiche

- $\emptyset$ e' una ER
- ε e' una ER

($\emptyset$ indica l'insieme vuoto, che e' diverso da ε, cioe' la parola vuota)

- se $e \in \Sigma$, e e' una ER

↳ cioe' un simbolo e' una ER

operatori su espressioni regolari

Se u e v sono ER:

- $u+v$ e' una ER

NOTA - $A+B \equiv A \cup B$ => UNIONE

- uv e' una ER => CONCATENAZIONE
- u^* e' una ER => CHIUSURA DI KLEENE

=> Le espressioni regolari permettono di rappresentare in forma compatta e finita linguaggi che contengono infinite parole.

ESEMPI - $\Sigma = \{a, b, c\}$

- $(a+b)c^*$

  $\Rightarrow \mathcal{L} = \{a, b, ac, bc, acc, bcc, \ldots\}$

- $(ab)^* + c$

  $\Rightarrow \mathcal{L} = \{\varepsilon, c, ab, abab, ababab, \ldots\}$

NOTA - Fra i tre operatori, l'operatore $*$ (chiusura di Kleene) ha la precedenza più alta, seguito dall'operatore $\bullet$ (concatenazione), e infine dall'operatore $+$ (unione). Le parentesi vengono usate per modificare l'ordine in cui gli operatori vengono applicati.

esempio - $ab^* + c \neq (ab)^* + c$

$ab^* + c \to \{c, a, ab, abb, \ldots\}$

$(ab)^* + c \to \{\varepsilon, c, ab, abab, \ldots\}$

DEFINIZIONE - Ogni linguaggio che può essere descritto da un'espressione regolare si dice LINGUAGGIO REGOLARE.

Si osservi che un linguaggio con un numero finito di parole è sempre un linguaggio regolare. Infatti, può essere descritto come "somma" (unione) delle sue parole (concatenazioni di simboli).

esempio - $\mathcal{L} = \{abc, bca, bc\} \Rightarrow abc + bca + bc$