

# PROGETTAZIONE

---

NICOLA SANITATE

## REDISCOVERING WORKFLOW MODELS FROM EVENT-BASED DATA

PROGETTO DI INTELLIGENZA ARTIFICIALE

A.A. 2010 - 2011

UNIVERSITÀ DEGLI STUDI DI BARI

---

# PROGETTAZIONE

## INDICE

<b>INTRODUZIONE.....</b>	<b>3</b>
<b>INPUT.....</b>	<b>4</b>
<b>OUTPUT.....</b>	<b>6</b>
<b>COSTRUZIONE DELLA D/F-TABLE.....</b>	<b>7</b>
<b>INDUZIONE DEL D/F-GRAPH.....</b>	<b>11</b>
<b>GENERAZIONE DELLA WF-NET.....</b>	<b>14</b>
<b>CONCLUSIONI.....</b>	<b>19</b>

---

# PROGETTAZIONE

## INTRODUZIONE

Questo documento ha lo scopo di concretizzare la tecnica presentata nel documento di analisi suddividendola in sottoproblemi al quale verranno associate delle soluzioni direttamente realizzabili in programmazione logica tramite linguaggio Prolog.

I primi problemi a cui si dovrà trovare soluzione sono:

- la lettura dell'input,
- la scrittura dell'output.

In questo modo si possono ben definire i confini della tecnica di Process Mining in quanto questa dovrà partire dai dati ottenuti dalla lettura dell'input e dovrà fornire i dati necessari alla scrittura dell'output.

In seguito quindi verrà effettuata una macro-suddivisione del problema di Process Mining seguendo le fasi in cui è suddivisa la tecnica al fine di agevolare il parallelismo tra la tecnica stessa e la sua implementazione:

- costruzione della D/F-table partendo da delle sequenze di task;
- induzione del D/F-graph partendo dalla D/F-table;
- generazione della WF-net a partire dalla D/F-table e dal D/F-graph.

In conclusione verrà effettuato l'accorpamento delle diverse parti e illustrati gli strumenti software che verranno utilizzati per lo sviluppo del sistema.

---

# PROGETTAZIONE

## INPUT

Il sistema dovrà accettare in input due elementi:

- la percentuale di errore attesa impostabile a run time,
- un file contenente le sequenze di task.

All'interno del file ogni riga corrisponde ad una sequenza di task formattata esattamente come specificato nell'articolo.

T1, T2, T4, T3, T5, T9, T6, T3, T5, T10, T8, T11, T12, T2, T4, T7, T3, T5, T8, T11, T13

Si è scelto di mantenere questo formalismo invece di sceglierne un altro che risulti di più facile gestione per Prolog per due motivi:

- coerenza con l'articolo
- necessità di scegliere un formalismo che sia facilmente ricavabile dai log dei sistemi informativi transazionali.

Dato il formalismo scelto si rende necessario realizzare un parser che associ:

- ad ogni gruppo di caratteri separato da una virgola una transizione (task),
- ad ogni gruppo di transizioni separato da un return una sequenza,
- all'intero gruppo di sequenze ricavate una lista di sequenze.

Per le transizioni si sceglie una rappresentazione mediante atomi, in quanto sono gli oggetti base per l'intero processo di Process Mining. Al contrario, sia per le sequenze che per la lista di sequenze si sceglie una rappresentazione sotto forma di lista in quanto ideale per rappresentare la dinamicità di questi oggetti.

Di seguito si riporta la soluzione trovata per la lettura del file di input.

Lettura di una transizione:

- termina se il carattere corrente è un return o end\_of\_file, e in tal caso restituisce 1 nel flag di fine riga e la lista di caratteri vuota;
- termina se il carattere corrente è una virgola, ma in tal caso restituisce 0 nel flag di fine riga e la lista di caratteri vuota;
- per gli altri casi preleva un nuovo carattere dallo stream di input e richiama la lettura della transizione sul nuovo carattere; alla lista dei caratteri ottenuta aggiungerà in testa il carattere corrente.

---

## PROGETTAZIONE

Lettura di una sequenza:

- termina se il flag di fine riga è uguale ad 1, e in tal caso restituisce la lista di transizioni vuota;
- per gli altri casi legge una transizione ottenendo una lista di caratteri, trasforma questa lista di caratteri in un atomo e richiama la lettura della sequenza; alla lista delle transizioni ottenuta aggiungerà in testa l'atomo ottenuto.

Lettura del file di input:

- termina se l'input stream raggiunge end\_of\_file, e in tal caso restituisci la lista delle sequenze vuota;
- per gli altri casi legge una riga ottenendo una sequenza di transizioni e richiama la lettura del file; alla lista delle sequenze ottenuta aggiungerà in testa la sequenza corrente.

A questo punto il processo di Process Mining potrà lavorare su una lista di sequenze di transizioni strutturate in questo modo:

`[[t1,t2,...],[t1,t2,...],...]`

## OUTPUT

Il sistema dovrà produrre in output tre elementi:

- dei messaggi a schermo che indichino la fase corrente affrontata dal sistema,
- un file che riporti la WF-net generata,
- un file di log che registri i risultati parziali dell'intero processo.

Per il primo file non ci sono indicazioni esplicite nell'articolo, e dato che questo file dovrà essere confrontato con un file simile contenente la rete originale durante la sessione di test, le opzioni individuate sono 3, tutte facilmente gestibili da Prolog:

1. `in(PostoPrecedente, Transizione), out(Transizione, PostoSuccessivo).`
2. `in([PostiPrecedenti], Transizione), out(Transizione,[PostiSuccessivi]).`
3. `in([[PostoPrecedente, Transizione],...]), out([[Transizione,PostoSuccessivo],...]).`

La scelta è da effettuare tra due insiemi di fatti raffiguranti gli archi in entrata ed uscita, due insiemi di fatti raffiguranti le liste di adiacenza per ogni transizione, e delle liste di archi. La scelta è ricaduta sul primo formalismo in quanto, garantisce una facile lettura ed una efficienza maggiore dovuta alla possibilità di accedere direttamente ad ogni arco ed ad ogni nodo.

Di seguito si riporta la soluzione trovata per la scrittura del primo file di output.

Scrittura della WF-net nel file di output:

- salva il vecchio stream di output, reindirizza lo stream di output verso il file di output, elenca tutti i fatti `in/2` e tutti i fatti `out/2`, chiude il file e reindirizza lo stream di output sul vecchio stream di output.

Questa soluzione richiede che al termine della generazione della WF-net tutti i fatti `in/2` e `out/2` inerenti al modello generato siano stati asseriti.

Anche per il secondo file non ci sono indicazioni esplicite nell'articolo, ma la scelta del formalismo cambierà a seconda della fase in cui verranno registrati i risultati parziali. Si lascia quindi la descrizione della scelta alle rispettive sezioni. Ad ogni modo è facile prevedere che il formalismo adottato dovrà avvantaggiare la lettura in quanto il file è destinato alla consultazione da parte degli analisti.

La soluzione trovata per questo problema si riduce all'apertura di uno stream di output all'inizio del processo di Process Mining, il passaggio di tale stream a tutti i sottoproblemi che ne hanno bisogno, e alla chiusura dello stesso al termine dell'intero processo.

---

# PROGETTAZIONE

## COSTRUZIONE DELLA D/F-TABLE

Per la costruzione della D/F-table si richiede, per ogni transizione in esame  $A$ , il calcolo della frequenza all'interno delle sequenze. In seguito la transizione sarà confrontata con le transizioni di riferimento  $B$  al fine di trovare, sempre all'interno delle sequenze, la frequenza di precedenze dirette, successioni dirette, precedenze, successioni e fattore di causalità come definito nel documento di analisi.

Per la rappresentazione della tabella le opzioni individuate sono 2:

1.  $[[A_1, \#A_1, [[B_1, B_1 < A_1, A_1 > B_1, B_1 < < < A_1, A_1 > > > B_1, A_1 \rightarrow B_1], [B_2, \dots], \dots], [A_2, \dots], \dots]]$ .
2.  $\text{riga}(A, \#A, [[B_1, B_1 < A, A > B_1, B_1 < < < A, A > > > B_1, A \rightarrow B_1], [B_2, \dots], \dots])$ .

La scelta è da effettuare tra una lista di righe al cui interno compaiono i dati sopra elencati inerenti ad una singola transizione, oppure un insieme di fatti raffiguranti anch'essi le righe della tabella. La scelta è ricaduta sul secondo formalismo in quanto si prevede la fruizione della tabella in modo selettivo anziché sequenziale; inoltre questa soluzione offre una maggiore efficienza fornita dalla possibilità di accesso diretto.

Si noti inoltre come, con una semplice inversione delle sequenze, i concetti di precedenza e successione, così come i concetti di precedenza diretta e successione diretta, diventino identici. Ciò significa che trovando una soluzione per il calcolo della successione e della successione diretta sulle sequenze nell'ordine iniziale, queste offriranno rispettivamente la precedenza e la precedenza diretta fornendogli le sequenze in ordine invertito.

Per il file di log in questa sezione si prevede la stampa delle tabelle ottenute tramite il seguente formalismo:

$A_1 = \$$

$\#A_1 = \$$

---

B	$B < A$	$A > B$	$B < < < A$	$A > > > B$	$A \rightarrow B$
---	---------	---------	-------------	-------------	-------------------

---

$B_1$	\$	\$	\$	\$	\$. \$
-------	----	----	----	----	--------

$B_2$	...	...	...	...	...
-------	-----	-----	-----	-----	-----

---

$A_2 = \$$

...

---

## PROGETTAZIONE

Di seguito si riporta la soluzione trovata per la costruzione della D/F-table.

Inversione delle sequenze:

- termina se non ci sono più sequenze da invertire, in tal caso restituisce la lista di sequenze invertite vuota;
- per gli altri casi inverte la sequenza corrente e richiama l'inversione delle sequenze sulle sequenze rimanenti; alla lista delle sequenze invertite ottenuta aggiungerà in testa la sequenza invertita corrente.

Calcolo della frequenza all'interno di una sequenza:

- termina se non ci sono più transizioni nella sequenza da analizzare, in tal caso restituisce 0 come frequenza corrente;
- se la transizione di cui si vuole conoscere la frequenza è in testa alla sequenza da analizzare allora richiama il calcolo della frequenza sul resto della sequenza; la frequenza ottenuta sarà incrementata di 1 e restituita come frequenza corrente;
- per gli altri casi richiama il calcolo della frequenza sulla sequenza escludendo la testa; la frequenza ottenuta sarà restituita come frequenza corrente.

Calcolo della frequenza nella lista di sequenze:

- termina se non ci sono più sequenze nella lista di sequenze, in tal caso restituisce 0 come frequenza corrente;
- per gli altri casi calcola la frequenza all'interno della sequenza di testa della lista delle sequenze e richiama il calcolo della frequenza sul resto della lista delle sequenze; la somma delle due frequenze ottenute sarà restituita come frequenza corrente.

Calcolo della successione diretta all'interno di una sequenza:

- termina se non ci sono più transizioni nella sequenza da analizzare, in tal caso restituisce 0 come successione diretta corrente;
- se la transizione in esame  $A$  è in testa alla sequenza da analizzare seguita dalla transizione di riferimento  $B$  allora richiama il calcolo della successione diretta sul resto della sequenza con  $B$  in testa; la successione diretta ottenuta sarà incrementata di 1 e restituita come successione diretta corrente;
- per gli altri casi richiama il calcolo della successione diretta sulla sequenza escludendo la testa; la successione diretta ottenuta sarà restituita come successione diretta corrente.



---

## PROGETTAZIONE

Calcolo della successione diretta nella lista di sequenze:

- termina se non ci sono più sequenze nella lista di sequenze, in tal caso restituisce 0 come successione diretta corrente;
- per gli altri casi calcola la successione diretta all'interno della sequenza di testa della lista delle sequenze e richiama il calcolo della successione diretta sul resto della lista delle sequenze; la somma delle due successioni dirette ottenute sarà restituita come successione diretta corrente.

Analisi della successione:

- termina se non ci sono più transizioni nella sequenza da analizzare o se la transizione in esame  $A$  è in testa alla sequenza da analizzare, in tal caso la successione non è valida quindi restituisce 0 come successione corrente e 0 come numero di transizioni intermedie;
- termina se la transizione di riferimento  $B$  è in testa alla sequenza da analizzare, in tal caso la successione è valida quindi restituisce 1 come successione corrente e 0 come numero di transizioni intermedie;
- per gli altri casi richiama l'analisi della successione sulla sequenza escludendo la testa; la successione ottenuta sarà restituita come successione corrente, mentre il numero di transizioni intermedie ottenuto sarà incrementato di 1 e restituito come numero di transizioni intermedie corrente.

Calcolo della successione all'interno di una sequenza:

- termina se non ci sono più transizioni nella sequenza da analizzare, in tal caso restituisce 0 come successione corrente e 0 come incremento alla causalità;
- se la transizione in esame  $A$  è in testa alla sequenza da analizzare allora richiede l'analisi della successione sul resto della sequenza e richiama il calcolo della successione sul resto della sequenza; la successione ottenuta dal resto della sequenza sarà incrementata della successione ottenuta dall'analisi (1 se la successione è valida e 0 altrimenti) e restituita come successione corrente, mentre l'incremento alla causalità ottenuta dal resto della sequenza sarà incrementato di un  $\delta^n \cdot s$ , dove  $\delta$  è fissato a 0.8,  $n$  è numero di transizioni intermedie ottenuto dall'analisi ed  $s$  è la successione (al fine di tener conto della validità della successione), e restituito come incremento alla causalità corrente;
- per gli altri casi richiama il calcolo della successione sulla sequenza escludendo la testa; la successione e l'incremento alla causalità ottenuti saranno restituiti rispettivamente come successione e incremento alla causalità correnti.

---

## PROGETTAZIONE

Calcolo della successione nella lista di sequenze:

- termina se non ci sono più sequenze nella lista di sequenze, in tal caso restituisce 0 come successione corrente;
- per gli altri casi calcola la successione all'interno della sequenza di testa della lista delle sequenze e richiama il calcolo della successione sul resto della lista delle sequenze; la somma delle due successioni e la somma dei due incrementi alla causalità ottenuti saranno restituiti rispettivamente come successione e incremento alla causalità corrente.

Costruzione della sottotabella:

- termina se non ci sono più transizioni nella lista delle transizioni distinte, in tal caso restituisce la lista di righe vuota;
- per gli altri casi, in relazione alla transizione di riferimento  $B$  in testa alla lista delle transizioni distinte, formerà una riga calcolando la precedenza diretta usando la successione diretta sulle sequenze invertite, la successione diretta, la precedenza usando la successione sulle sequenze invertite, la successione e la causalità, sottraendo l'incremento alla causalità della precedenza all'incremento alla causalità della successione, e dividendo il tutto per la frequenza complessiva della transizione in esame  $A$ ; in seguito stampa sul file di log la riga appena ottenuta; infine richiama la costruzione della sottotabella sulle transizioni distinte rimanenti; alla lista di righe ottenuta aggiungerà in testa la riga corrente.

Costruzione della D/F-table:

- termina se non ci sono più transizioni nella lista delle transizioni distinte;
- per gli altri casi, in relazione alla transizione in esame  $A$  in testa alla lista delle transizioni distinte, asserisce una nuova riga della D/F-table calcolando la frequenza e la sottotabella; stampa su file di log la frequenza ottenuta e la struttura per la sottotabella; infine richiama la costruzione della tabella sulle transizioni distinte rimanenti.

## INDUZIONE DEL D/F-GRAPH

Per l'induzione del D/F-graph si richiede, per ogni transizione in esame  $A$  (ora anche nodo del grafo), un'analisi della riga corrispondente nella D/F-table al fine di individuare, sulla base delle tre euristiche definite nel documento di analisi, delle connessioni di causalità (interpretati come archi del grafo) tra la transizione e altre transizioni di riferimento  $B$ .

Per la rappresentazione del grafo le opzioni individuate sono 3:

1. arco( $A, B$ ).
2. archi( $A, [B_1, B_2, \dots]$ ).
3.  $[[A_1, B_1], [A_2, B_2], \dots]$ .

La scelta è da effettuare tra un insieme di fatti raffiguranti gli archi, un insieme di fatti raffiguranti la liste di adiacenza per ogni nodo, e una lista di archi. La scelta è ricaduta sul primo formalismo in quanto si prevede la fruizione del grafo in modo selettivo anziché sequenziale; inoltre, come già detto, questa soluzione offre una maggiore efficienza fornita dalla possibilità di accesso diretto.

Seguendo la definizione di differenza relativa

$$\text{differenza relativa}(x, y) = \left| \frac{x - y}{\frac{x + y}{2}} \right| ,$$

si mostra l'interpretazione data alle similitudini presenti nelle euristiche 2 e 3 (considerando che si tratta di valori positivi è possibile evitare di eseguire eventuali divisioni per 0 moltiplicando ambo i membri per il valore del denominatore).

- Euristica 2:

$$(A \rightarrow A \approx 0) \equiv \left( |A \rightarrow A| \leq N \cdot \left| \frac{A \rightarrow A}{2} \right| \right) .$$
$$(A < A - A > A \approx 0) \equiv \left( |A < A - A > A| \leq N \cdot \left| \frac{A < A - A > A}{2} \right| \right) .$$

- Euristica 3:

$$(A \rightarrow B \approx 0) \equiv \left( |A \rightarrow B| \leq N \cdot \left| \frac{A \rightarrow B}{2} \right| \right) .$$
$$(A < B \approx A > B) \equiv \left( |A < B - A > B| \leq N \cdot \left| \frac{A < B + A > B}{2} \right| \right) .$$
$$(A < < < B \approx A > > > B) \equiv \left( |A < < < B - A > > > B| \leq N \cdot \left| \frac{A < < < B + A > > > B}{2} \right| \right) .$$

---

# PROGETTAZIONE

Per il file di log in questa sezione si prevede la stampa degli archi ottenuti seguiti dalla euristica che ne ha permesso l'individuazione seguendo il seguente formalismo:

$[A,B] \rightarrow \text{euristica } S$

Di seguito si riporta la soluzione trovata per l'induzione del D/F-graph.

Euristica 1:

- controlla che
  - la causalità sia maggiore o uguale al fattore di rumore,
  - la successione diretta sia maggiore o uguale al valore di soglia,
  - la precedenza diretta sia minore o uguale al valore di soglia;

se i tre controlli vanno a buon fine asserisce un nuovo arco del D/F-graph uscente dal nodo in esame  $A$  ed entrante nel nodo di riferimento  $B$ ; stampa su file di log il nuovo arco trovato associando la scoperta ad Euristica 1.

Euristica 2:

- controlla che
  - la causalità in valore assoluto sia minore o uguale al prodotto tra fattore di rumore e il 50% della causalità in valore assoluto,
  - la somma tra precedenza diretta e successione diretta sia maggiore del 50% della frequenza,
  - la sottrazione tra precedenza diretta e successione diretta in valore assoluto sia minore o uguale al prodotto tra il fattore di rumore ed il 50% della sottrazione tra precedenza diretta e successione diretta in valore assoluto;

se i tre controlli vanno a buon fine asserisce un nuovo arco del D/F-graph ricorsivo sul nodo in esame  $A$ ; stampa su file di log il nuovo arco trovato associando la scoperta ad Euristica 2.

Euristica 3:

- controlla che
  - la causalità in valore assoluto sia minore o uguale al prodotto tra fattore di rumore e il 50% della causalità in valore assoluto,
  - la successione diretta sia maggiore o uguale al valore di soglia,
  - la differenza tra precedenza diretta e successione diretta in valore assoluto sia minore o uguale al prodotto tra il fattore di rumore e la media aritmetica tra precedenza diretta e successione diretta,

---

## PROGETTAZIONE

- la successione sia maggiore o uguale del 40% della frequenza del nodo in esame  $A$ ,
- la differenza tra precedenza e successione in valore assoluto sia minore o uguale al prodotto tra il fattore di rumore e la media aritmetica tra precedenza e successione;

se i cinque controlli vanno a buon fine asserisce un nuovo arco del D/F-graph uscente dal nodo in esame  $A$  ed entrante nel nodo di riferimento  $B$ ; stampa su file di log il nuovo arco trovato associando la scoperta ad Euristiche 3.

Applicazione dell'euristica:

- se la transizione di riferimento  $B$  è uguale alla transizione in esame  $A$ , allora cerca di applicare l'euristica 2;
- per gli altri casi cerca di applicare l'euristica 1;
- se l'euristica 1 fallisce allora cerca di applicare l'euristica 3.

Induzione della parte del grafo inerente ad una transizione:

- termina se non ci sono più righe nella sottotabella inerente alla transizione;
- per gli altri casi cerca di applicare un'euristica tra la transizione in esame  $A$  e la transizione di riferimento  $B$  prelevata dalla riga corrente della sottotabella; in seguito richiama l'induzione della parte del grafo inerente ad una transizione sulle righe rimanenti;
- se il passo precedente fallisce non riuscendo ad applicare alcuna euristica, richiama l'induzione della parte del grafo inerente ad una transizione sulle righe della sottotabella escludendo la testa.

Induzione del D/F-graph:

- termina se non ci sono più transizioni nella lista delle transizioni distinte;
- per gli altri casi ottiene la riga della D/F-table inerente alla transizione in esame  $A$  in testa alla lista delle transizioni distinte, elabora l'induzione della parte di grafo inerente alla transizione in esame  $A$  e richiama l'induzione del D/F-graph sulle transizioni distinte rimanenti.

## GENERAZIONE DELLA WF-NET

Per la generazione della WF-net si richiede, per ogni transizione in esame  $A$  un'analisi degli archi del D/F-graph sia entranti che uscenti al fine di individuare, sulla base dei dati presenti nella D/F-table, la tipologia di join e split. Il documento di analisi presentava tre modi equivalenti per ottenere la giusta tipologia di join e split, ma si sceglie di utilizzare la prima in modo da riutilizzare l'algoritmo di esempio presentato in pseudo-codice.

L'intuizione alla base del modo scelto è la seguente: delle precedenze dirette e successioni dirette frequenti di due o più task di destinazione (partenza) di uno split (join) sono indice del fatto che questi task vengano eseguiti in modo parallelo, e quindi è presumibile che si sia in presenza di un AND-split (AND-join). Al contrario, delle precedenze dirette e successioni dirette poco frequenti sono indice del fatto che questi task vengano eseguiti in modo opzionale esclusivo, e quindi è presumibile che si sia in presenza di un OR-split (OR-join).

Per la rappresentazione della WF-net si è già trattato nella sezione “Output”. Per la rappresentazione degli split e dei join invece, le opzioni individuate sono 2:

1.  $\text{split}(A, [[B_1, B_2], [B_3, \dots], \dots]), \text{join}(A, [[B_1, B_2], [B_3, \dots], \dots])$ .
2.  $\text{split}([A_1, [[B_1, B_2], [B_3, \dots], \dots]], [A_2, [[B_1, B_2], [B_3, \dots], \dots]]), \text{join}([A_1, [[B_1, B_2], [B_3, \dots], \dots]], [A_2, [[B_1, B_2], [B_3, \dots], \dots]])$ .

La scelta è da effettuare tra un insieme di fatti raffiguranti split e join rispetto ad una singola transizione e due liste, una contenente split e l'altra contenente join di tutte le transizioni. La scelta è ricaduta sul primo formalismo in quanto, ancora una volta, questa soluzione offre una maggiore efficienza fornita dalla possibilità di accesso diretto.

Per il file di log in questa sezione si prevede la stampa delle connessioni ottenute in forma preposizionale:

$A \quad (B_1 \text{ OR } B_2) \text{ AND } (B_3 \text{ OR } B_4)$

Di seguito si riporta la soluzione trovata per la generazione della WF-net.

Ricerca della successione diretta relativa ad una transizione in una sottotabella:

- termina se la riga della sottotabella corrisponde alla transizione di riferimento  $B$ , e in tal caso restituisce la successione diretta trovata nella riga;
- per gli altri casi richiama la ricerca della successione diretta sulla sottotabella escludendo la testa; la successione diretta ottenuta sarà la successione diretta corrente.

---

## PROGETTAZIONE

Valutazione dei requisiti di inserimento di una transizioni in un insieme:

- termina se non ci sono più elementi nell'insieme di riferimento;
- per gli altri casi ottiene la riga della D/F-table inerente alla transizione in esame  $A$ , preleva la successione diretta tra la transizione in esame  $A$  ed il primo elemento  $B$  dell'insieme di riferimento e valuta che questa sia minore del valore di soglia (se fallisce la disuguaglianza deve fallire l'intera valutazione in quanto la ricerca della successione diretta non fornirebbe altri valori validi); in seguito ottiene la riga della D/F-table inerente al primo elemento  $B$  dell'insieme di riferimento, preleva la successione diretta tra l'elemento  $B$  e la transizione in esame  $A$  e valuta che questa sia minore del valore di soglia (per i motivi mostrati in precedenza, anche in questo caso se fallisce la disuguaglianza deve fallire l'intera valutazione); infine richiama la valutazione tra la transizione in esame  $A$  e gli elementi rimanenti.

Collocazione di una transizioni nell'insieme più opportuno:

- termina se non ci sono più insiemi nella lista degli insiemi, e in tal caso restituisce una lista degli insiemi contenente un nuovo insieme, contenente a sua volta la transizione in esame  $A$ ;
- per gli altri casi cerca di valutare se la transizione in esame  $A$  possiede i requisiti per essere integrato nell'insieme di testa della lista degli insiemi; in tal caso aggiunge la transizione in esame  $A$  all'insieme di testa della lista degli insiemi;
- se il passo precedente fallisce perché la valutazione non è andata a buon fine, richiama la collocazione della transizione in esame  $A$  sugli insiemi della lista degli insiemi escludendo l'insieme di testa.

Costruzione degli insiemi:

- termina se non ci sono più transizioni nella lista delle transizioni distinte, e in tal caso restituisce una lista degli insiemi vuota;
- per gli altri casi richiama la costruzione degli insiemi sulla lista delle transizioni distinte escludendo la testa, e in seguito aggiorna e restituisce la lista degli insiemi ottenuta collocando la transizione di testa nell'insieme più opportuno.

Stampa di un sottoinsieme:

- se il sottoinsieme contiene un solo elemento allora stampa quell'elemento;
- per gli altri casi stampa il primo elemento del sottoinsieme e la scritta "OR", e in seguito richiama la stampa del sottoinsieme sugli elementi rimanenti.

---

## PROGETTAZIONE

Stampa di un insieme:

- se l'insieme è vuoto allora stampa due parentesi, una aperta ed una chiusa;
- se l'insieme contiene un solo sottoinsieme allora stampa una parentesi aperta, il sottoinsieme, e una parentesi chiusa;
- per gli altri casi stampa una parentesi aperta, il primo sottoinsieme dell'insieme, una parentesi chiusa e la scritta “AND”, e in seguito richiama la stampa dell'insieme sui sottoinsiemi rimanenti.

Classificazione delle divisioni (split):

- termina se non ci sono più transizioni nella lista delle transizioni distinte;
- per gli altri casi ottiene una lista di tutte le transizioni di destinazione degli archi che hanno come transizione di partenza la transizione di testa della lista delle transizioni distinte, divide la lista ottenuta costruendo degli insiemi e asserisce un nuovo split contenente la transizione in esame  $A$  e gli insiemi ottenuti; in seguito stampa su file di log la transizione in esame  $A$  e gli insiemi ottenuti, e richiama la classificazione delle divisioni sulle transizioni distinte rimanenti.

Classificazione delle unioni (join):

- termina se non ci sono più transizioni nella lista delle transizioni distinte;
- per gli altri casi ottiene una lista di tutte le transizioni di partenza degli archi che hanno come transizione di destinazione la transizione di testa della lista delle transizioni distinte, divide la lista ottenuta costruendo degli insiemi e asserisce un nuovo join contenente la transizione in esame  $A$  e gli insiemi ottenuti; in seguito stampa su file di log la transizione in esame  $A$  e gli insiemi ottenuti, e richiama la classificazione delle unioni sulle transizioni distinte rimanenti.

Asserzione di archi da un insieme di transizioni ad un posto:

- termina se non ci sono più transizioni nell'insieme di transizioni;
- per gli altri casi asserisce un nuovo out tra la prima transizione dell'insieme di transizioni ed il posto e richiama l'asserzione di archi da un insieme di transizioni ad un posto sulle transizioni rimanenti.

Asserzione di archi da un posto ad un insieme di transizioni:

- termina se non ci sono più transizioni nell'insieme di transizioni;
- per gli altri casi asserisce un nuovo in tra il posto e la prima transizione dell'insieme di transizioni e richiama l'asserzione di archi da un posto ad un insieme di transizioni sulle transizioni rimanenti.



---

## PROGETTAZIONE

Ricerca del sottoinsieme in cui compare una transizione:

- se il sottoinsieme in testa alla lista di sottoinsiemi contiene la transizione in esame, allora restituisce il sottoinsieme in questione;
- per gli altri casi richiama la ricerca del sottoinsieme sui sottoinsiemi rimanenti.

Asserzione dei confini di un posto:

- partendo da una transizione di partenza ed una lista di transizioni di destinazione di un OR-split, ottiene la lista di sottoinsiemi di partenza di una join relativi alla transizione di testa della lista, da questa ricava il sottoinsieme in cui compare la transizione di partenza; in questo modo viene a conoscenza di tutte le transizioni precedenti e successive ad un determinato posto, dunque asserisce prima gli archi dall'insieme di transizioni precedenti verso il posto e poi gli archi dal posto all'insieme di transizioni successive.

Valutazione di una divisione:

- termina se non ci sono più sottoinsiemi nella lista delle divisioni di una transizione, e in tal caso restituisce come nuovo numero di posto il numero di posto corrente;
- se riesce ad ottenere un posto dalla transizione in esame tramite un arco out, e da questo riesce ad ottenere la prima transizione dal primo sottoinsieme di divisioni tramite un arco in, vuol dire che il sottoinsieme è già stato analizzato ed asserito, dunque richiama la valutazione della divisione sui sottoinsiemi rimanenti senza variazioni per il nuovo numero di posto;
- per gli altri casi trasforma il numero di posto in una lista di caratteri, ottiene un atomo il cui nome corrisponde da una “p” seguita dalla lista appena ottenuta, incrementa il numero di posto di 1, asserisce i confini del posto appena creato e richiama la valutazione della divisione sui sottoinsiemi rimanenti passando il numero di posto aggiornato.

Asserzioni degli archi:

- termina se non ci sono più transizioni nella lista delle transizioni distinte;
- per gli altri casi ottiene l'insieme delle divisioni inerenti alla transizione di testa della lista delle transizioni distinte, valuta le divisioni ottenendo il numero di posto ancora inutilizzato e richiama l'asserzione degli archi sulle transizioni distinte rimanenti con il numero di posto ottenuto.

---

## PROGETTAZIONE

Asserzione degli archi iniziali:

- termina se non ci sono più transizioni nella lista di transizioni di destinazione;
- per gli altri casi asserisce un nuovo arco in dal posto “sb” alla transizione di testa della lista di transizioni di destinazione e richiama l'asserzione degli archi iniziali sulle transizioni di destinazione rimanenti.

Asserzione degli archi finali:

- termina se non ci sono più transizioni nella lista di transizioni di partenza;
- per gli altri casi asserisce un nuovo arco out dalla transizione di testa della lista di transizioni di destinazione al posto “se” e richiama l'asserzione degli archi finali sulle transizioni di partenza rimanenti.

Generazione della WF-net:

- scrive l'intestazione degli split sul file di log, classifica le divisioni, scrive l'intestazione dei join sul file di log, classifica le unioni, ottiene le transizione connesse al posto iniziale elencando le transizioni dei join che presentano insieme vuoti, asserisce gli archi iniziali, ottiene le transizione connesse al posto finale elencando le transizioni degli split che presentano insieme vuoti, asserisce gli archi finali e infine asserisce tutte gli archi rimanenti.

---

# PROGETTAZIONE

## CONCLUSIONI

Il problema di Process Mining è stato risolto in tutte le sue componenti. A questo punto serve un gestore che permetta di organizzare le soluzioni ai sottoproblemi mostrate in precedenza. Questo gestore avrà anche il compito di chiedere all'utente i dati in input da inserire a run time, ossia la percentuale di errore da gestire, e le stampe a schermo di messaggi che indichino lo stato di avanzamento del sistema nelle varie fasi di risoluzione del problema.

Per quanto riguarda i file di input e di output si sceglie di mantenere dei nomi fissi al fine di non caricare l'utente di troppe richieste, perdendo così un tempo maggiore di quello speso per la reale esecuzione del sistema. In particolare:

- workflow.log è il file contenente tutte le sequenze di workflow di input;
- processmining.log è il file contenente tutti i risultati intermedi del processo;
- wf-net-out è il file contenente la WF-net generata dal sistema.

Di seguito si riporta la soluzione trovata per il gestore iniziale. Data la mole di funzioni da svolgere, questa soluzione verrà presentata con un formalismo diverso al fine di agevolare la lettura.

Start:

- chiede all'utente di inserire la percentuale di rumore che si vuole gestire nella valutazione delle sequenze di input,
- attende la risposta dell'utente,
- ottiene il fattore di rumore dividendo la percentuale ottenuta per 100,
- esegue la lettura delle sequenze aprendo uno stream in lettura che punta al file “workflow.log” che verrà chiuso subito dopo,
- ottiene la lista delle transizioni distinte dall'insieme delle sequenze ottenute,
- calcola il valore di soglia sommando ad 1 il rapporto arrotondato del prodotto del numero di sequenze per il fattore di rumore sul numero di transizioni distinte,
- ricava un insieme di sequenze inverse rispetto a quelle ottenute precedentemente,
- apre uno stream in scrittura che punta al file “processmining.log”,
- genera la D/F-table scrivendo sia a schermo che sullo stream di output che si sta entrando nella fase di costruzione della D/F-table,
- genera il D/F-graph scrivendo sia a schermo che sullo stream di output che si sta entrando nella fase di induzione del D/F-graph,

---

## PROGETTAZIONE

- genera la WF-net scrivendo sia a schermo che sullo stream di output che si sta entrando nella fase di generazione della WF-net,
- chiude lo stream di output,
- esegue la scrittura della WF-net nel file “wf-net-out”,
- cancella tutto ciò che è stato asserito durante l'esecuzione del sistema per permettere ulteriori esecuzioni successive.

Per l'implementazione del sistema si è scelto di utilizzare degli strumenti che agevolino la scrittura di programmi in linguaggio Prolog. In particolare saranno utilizzati:

- Yap nella versione 6.2.2 come interprete Prolog;
- Gedit nella versione 3.2.1 per la stesura del codice in quanto fornisce uno schema di colori dedicato al Prolog che facilita di molto la distinzione dei diversi elementi;
- la libreria standard “lists”, che fornisce alcuni strumenti che facilitano la gestione delle liste;

Per la realizzazione della documentazione invece si fa uso di strumenti quali:

- LibreOffice nella versione 3.4.3 per la stesura dei documenti;
- Dia nella versione 0.97.1 per la realizzazione dei grafici;
- PrologDoc nella versione 1.0RC2 per l'estrapolazione automatica della documentazione di implementazione in formato HTML.
- HtmlDoc nella versione 1.8.27 per la realizzazione di una versione stampabile della documentazione di implementazione.

Per la documentazione di implementazione verrà utilizzato il formalismo adottato dalla documentazione di Yap.