Il progetto è stato fatto con un grafo:

*//----------GRAPH-------------*

typedef struct Graph {

    hashtable\* nodes; *//Nodekey -> GraphNode*

    hashtable\* forest; *//Relation -> NodeSetByRel*

    vector\_string\* relations;

} Graph;

che contiene dei puntatori ad hashtable

typedef struct hashtable

{

    unsigned size;

    unsigned capacity;

    LIST\* data;

} hashtable;

e ogni hashtable contiene una lista di dati partendo da LIST

typedef LNODE\* LIST;

typedef struct LNODE {

    char\* key;

    void\* value;

    struct LNODE\* next;

    struct LNODE\* prev;

} LNODE;

Con LIST che è un puntatore a una lista di nodi

Tutto questo lo useremo per gestire i nostri nodi ident, il value void per quanto riguarda la gestione dei nodi entità sarà un graphnode

typedef struct GraphNode

{

    char\* key;

    hashtable\* out; *//NodeKey -> vector<Relation>*

    hashtable\* in; *//NodeKey -> vector<Relation>*

    hashtable\* in\_per\_rel; *//Relation -> int*

} GraphNode;

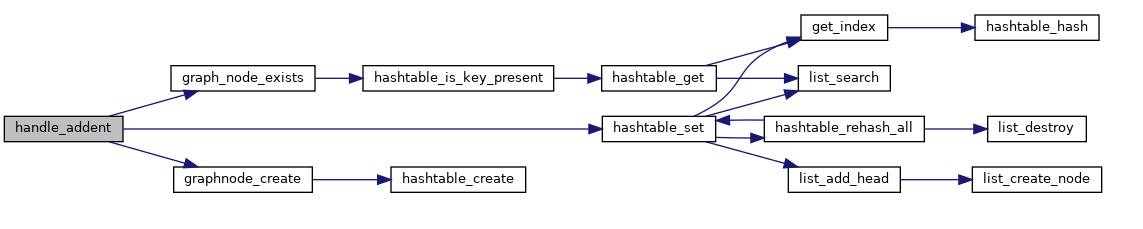
- Foresta di alberi delle relazioni entranti, dove ogni albero è legato a una relazione e le varie foglie sono i nodi della hashtable con valore un contatore delle relazioni entranti (del tipo dell’albero logicamente).

- Vettore di relazioni.

1) creazione di un grafo G

2) lettura dei comandi:

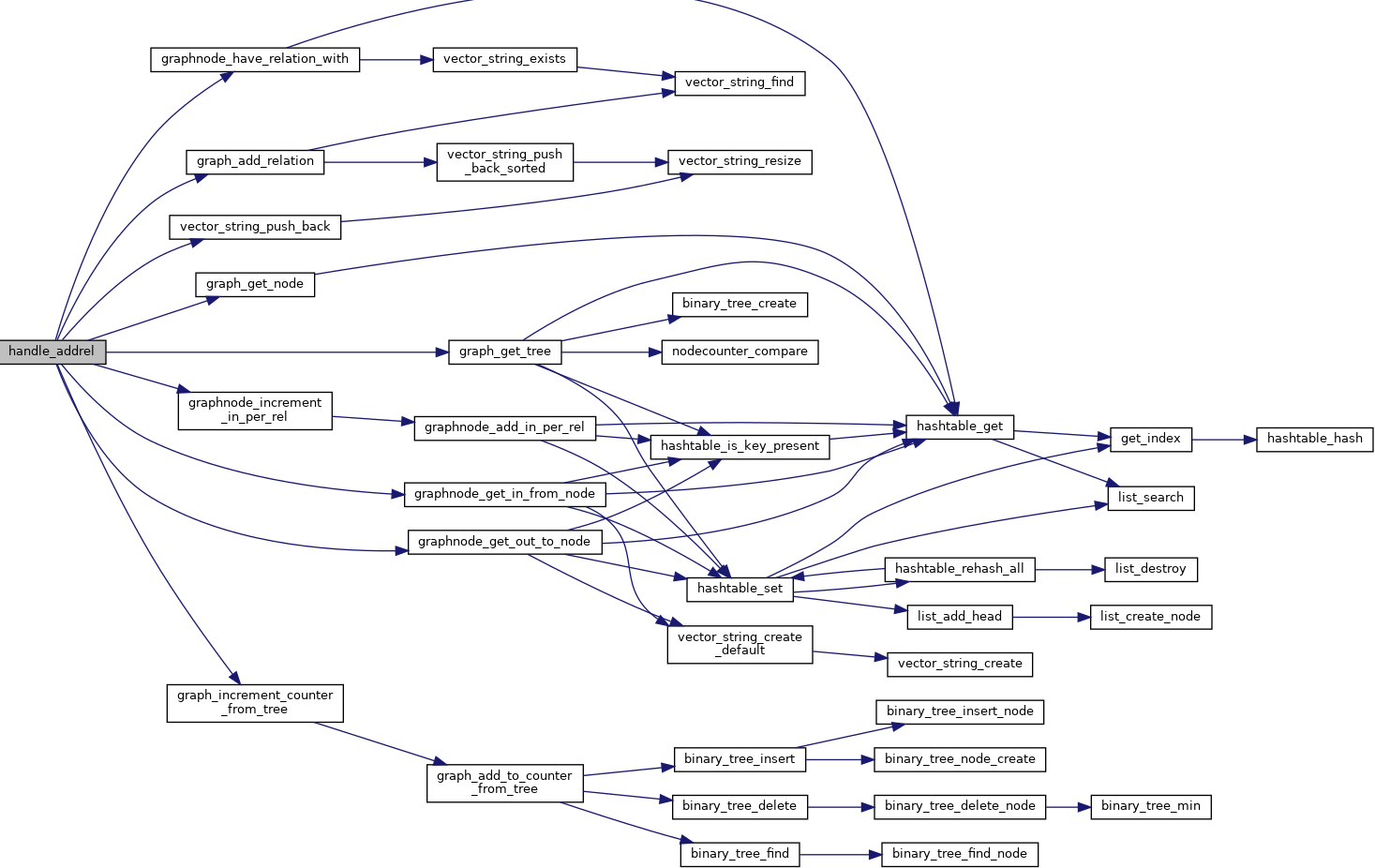
A) Add\_ent



Prima controllo se esiste già un nodo nel grafo con nome uguale al nome della nuova entità che devo aggiungere (cerco nella hashtable “nodes” che in pratica è l’hashtable di tutti i nodi ident e dopo aver ottenuto l’indice (possibile posizione iniziale ident) come per le hashtable partendo dall’indice LIST list\_of\_key = t->data[index] cerco poi se è proprio ident o se devo viaggiarmi il resto della lista con LIST node = list\_search(list\_of\_key, key);)

se c’è: non faccio nulla,

se manca: prima creo il nodo col nome poi lo aggiungo in testa alla lista di altri nodi, dopo aver controllato se il nodo esiste già e se va ampliata o meno la hashtable.

B) Add\_rel 

Prima di tutto controllo se i nodi coi nomi esistono e se manca la relazione cercando nel vettore di stringhe (contenuto nel nodo nella sezione out) se contiene la relazione.

Nel caso creo la relazione con idrel=“graph\_add\_relation(g,idrel)” (e se non c’è la aggiungo al vector\_string di relazioni) e poi creo dei puntatori a un vettore di stringhe che indicano l’esistenza della relazione tra il nodo 1 e il nodo 2 entrante per n2 e uscente per n1. Poi con “vector\_string\_push\_back” inserisco il nome della relazione ai due puntatori di prima.

Infine, vado a prendere il contatore corrente di relazioni entranti in n2 per la relazione idrel (che se è una relazione nuova deve essere aumentato da graphnode\_increment\_in\_per\_rel), prendo l’albero della relazione idrel e vado a cercare la foglia di questo albero con nome ident2 e inserisco il nuovo contatore.