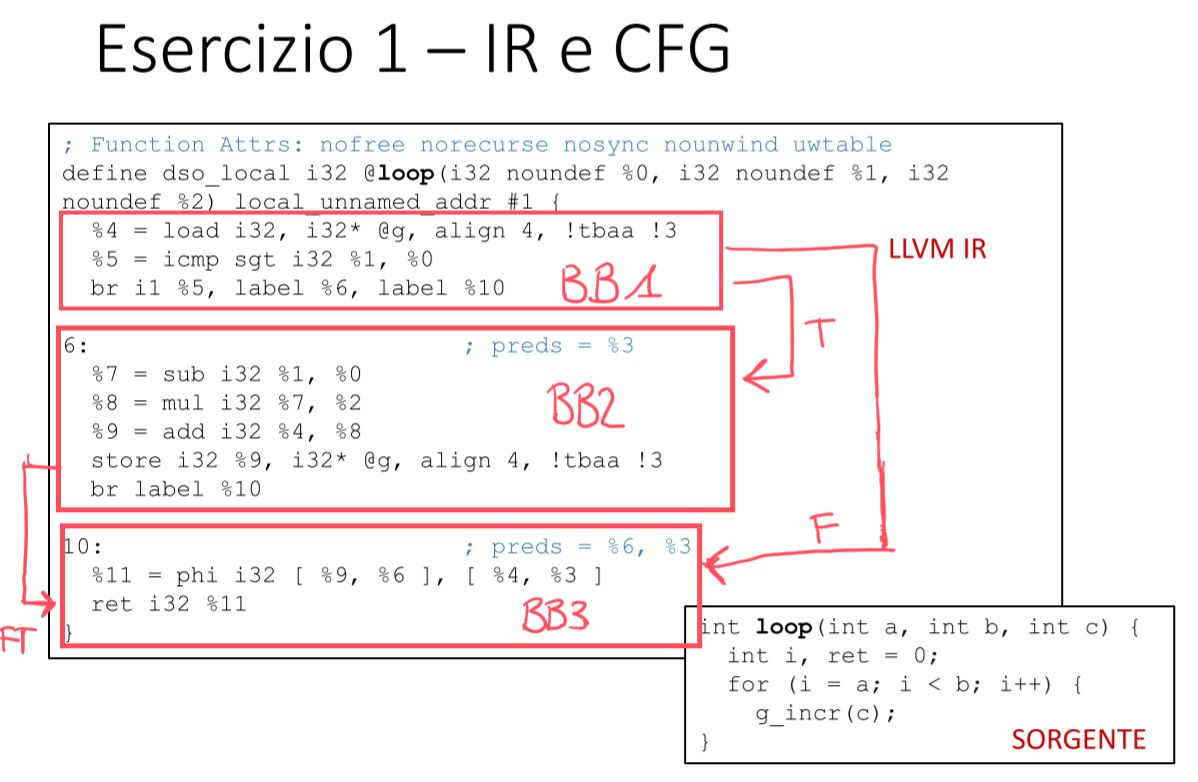
**ESERCIZIO 1-2 Gabriele Cusato 152669 , Alessandro Giacconi 152533**

Disegnare il CFG per ogni funzione:

Immagine che contiene testo

Descrizione generata automaticamente



Cos’è successo al loop:

Il loop non c’è più perché sono state apportate delle ottimizzazioni alla IR e questo si può notare dalla label ‘6’ dove avviane una moltiplicazione che serve ad eliminare appunto il loop.

Immagine che contiene testo

Descrizione generata automaticamente

Il tag O0 non effettua ottimizzazioni, traducendo alla lettera il codice in IR, mentre O2 ottimizza rimuovendo il loop.

Per esempio col flag -O2 i registri vengono prima tutti allocati e poi viene effettuata la store, invece col flag -O2 i registri vengono ottimizzati. L'opzione "-Rpass" del comando clang viene utilizzata per generare report dettagliati sull'applicazione dei pass di ottimizzazione sul codice sorgente.

(Esercizio 2) Dopo aver creato la rappresentazione intermedia anche per il modulo di Fibonacci, si è completato il file TestPass.cpp con i controlli e gli output richiesti:

Immagine che contiene testo, interno, schermata

Descrizione generata automaticamente

Tramite il comando:

opt -load-pass-plugin=./libTestPass.so -passes=test-pass test/<nome-file>.ll -disable-output

è possibile testare il file TestPass e visualizzare l’analisi dei vari moduli

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Il nome della funzione è stato ottenuto tramite il metodo “getName” della funzione F, ottenuta iterando il modulo passato come parametro al comando “opt” scritto sopra. Per ottenere il valore in stringa stampabile è stato usato il costrutto “.data()”.

Per stampare il numero di argomenti è stato usato il metodo getFunctionType() e “getNumParams”, mentre per identificare le funzioni variadiche si è utilizzato il metodo “getFunctionType”

Successivamente è stata usata una serie di cicli for per ciclare le altre funzioni del modulo, insieme ai basic blocks e le singole istruzioni, sia per contare i vari dati richiesti nella consegna, sia per verificare il numero di chiamate della funzione specifica analizzata. Infatti dentro il for più interno sono presenti 2 costrutti ‘if’ per controllare se l’istruzione analizzata è una chiamata della funzione analizzata.

Infine sono stati stampati i valori dei counter definiti in precedenza.