

assignment 4 : loop fusion

- obiettivo: sfruttare meglio il data reuse del programma, avvicinando accessi alle stesse locazioni che si trovano in punti diversi del programma
- ovvero: c'è data reuse nel programma, ma al momento non riesco ad usarlo

struttura del passo

- serve un `FunctionPass`
- vedremo poi come organizzare l'iterazione sui loop della funzione: in particolare la questione si complica in caso di loop innestati, dunque useremo un vettore `worklist`?

algoritmo

condizioni per la loop fusion:

1. i due loop devono essere adiacenti: non ci devono essere statement tra la fine del primo e l'inizio del secondo
2. devono iterare lo stesso numero di volte
3. devono essere **control flow equivalent**: devo garantire che entrambi eseguano (se eseguono)
4. non devono esistere dipendenze a distanza negativa tra il primo loop e il secondo:
 - una dip a distanza negativa avviene quando nel secondo loop a iterazione m viene usato un valore calcolato nel primo loop a iterazione $m+n$ con $n > 0$

adiacenza dei loop

devo evidentemente controllare l'uguaglianza tra exit bb e header ? recupera da dani

unica accortezza: se il loop è guarded (rivedi terminologia loop llvm da pagina docu) → controllo molto semplice da svolgere

3 control flow equivalenza

affinche si possa fare la funzione devo garantire che l'esecuzione di un loop implichi l'esecuzione anche dell'altro, **in entrambe le direzioni!!**

per fare il controllo verifico nuovamente condizioni di dominanza:

- evidente per la prima condizione (primo loop domina il secondo significa che il secondo deve eseguire per forza se esegue il primo)
- per la seconda: devo controllare le condizioni di **postdominanza** (recupero tramite `AM.getResult<PostDominatorTreeAnalysis>(F)`)

ovviamente devo includere header file necessari