

---

Il mio progetto di tesi ha come obiettivo lo studio e la modellizzazione delle strutture gerarchiche ereditarie prodotte nella programmazione a oggetti.

L'orientamento agli oggetti è un *paradigma di programmazione* volto a permettere un migliore e più organico riutilizzo del codice. Nei linguaggi di programmazione *object-oriented* è infatti possibile la dichiarazione di speciali strutture che prendono il nome di *oggetti*, contenenti dati e metodi e aventi la possibilità di interagire e condividere parti di codice. La condivisione di codice in particolare viene resa possibile attraverso il meccanismo dell'ereditarietà: un oggetto ha la possibilità di *ereditare* e utilizzare tutto o parte del codice presente in un altro oggetto precedentemente costruito. L'ereditarietà permette la costruzione di strutture gerarchiche complesse, formalmente rappresentabili con grafi aciclici direzionati.

Il mio lavoro inizia con la ricostruzione dei dati delle gerarchie, attraverso il download di circa 70000 pacchetti di codice scritti in tre diversi linguaggi di programmazione a oggetti (C++, Java e Python) dal portale GitHub, il più grande database online di codice sorgente. Sono stati inoltre realizzati diversi programmi per la ricostruzione delle strutture gerarchiche ereditarie, ottenute direttamente dal codice sorgente.

In questa tesi vengono proposti diversi modelli per l'interpretazione dei risultati ottenuti dall'analisi dei dati, con l'intento di cogliere le motivazioni fondamentali del meccanismo dell'ereditarietà nella programmazione a oggetti. Per semplicità, si sono scelti modelli che idealizzano il concetto di *albero ereditario*, un sottocaso del grafo aciclico direzionato nel quale non si tiene conto dell'ereditarietà multipla.

Un primo approccio alla modellizzazione del sistema viene data attraverso un modello *mean field* che tiene conto della competizione tra il vantaggio ottenuto dalla costruzione di strutture gerarchiche e la difficoltà che più oggetti diversi possano utilizzare molto codice in comune e dividerlo. Questo modello propone un meccanismo di ottimizzazione delle strutture gerarchiche fondato sulla minimizzazione di un funzionale che tiene conto di tale competizione.

Per modellizzare più in dettaglio la topologia della struttura gerarchica dei grafi ho introdotto il modello *Sharing Tree model*, naturale estensione del modello mean field basata esplicitamente su calcoli combinatori. Questo modello permette una costruzione diretta di strutture gerarchiche attraverso simulazioni Monte Carlo ed è in buona parte risolto analiticamente.

L'analisi dei dati è stata eseguita attraverso la costruzione di una libreria

---

in C++ ed alcuni programmi in C, gvpr e bash. Dall'analisi dati sono state estratte molte informazioni interessanti, evidenziati alcuni comportamenti universali nei linguaggi di programmazione a oggetti, ma anche messe in luce alcune differenze strutturali nella gestione dell'ereditarietà.

Il numero di oggetti che compongono le strutture gerarchiche ha una distribuzione a potenza, sostanzialmente universale rispetto ai tre linguaggi di programmazione analizzati. Questa forma nella distribuzione delle *taglie* delle strutture gerarchiche è causa della medesima forma nella distribuzione di molte altre quantità: la profondità delle strutture (il numero massimo di oggetti che separa l'oggetto radice dell'albero dai rami finali della gerarchia) e il numero di *outdegree* (numero di oggetti che ereditano il codice) di ogni oggetto hanno anch'essi una distribuzione a potenza.

In particolare, il linguaggio di programmazione Java presenta strutture sistematicamente meno profonde rispetto agli altri due linguaggi presentati. Questo ed altri comportamenti anomali sono frutto di alcune importanti differenze nella gestione dell'ereditarietà decise dagli ideatori del linguaggio Java e con l'obiettivo di minimizzare la complessità strutturale del codice, obiettivo che grazie all'analisi dati possiamo dichiarare raggiunto sotto molti aspetti.

L'osservazione critica dei dati attraverso le previsioni del modello mean field propone un primo risultato interessante: le gerarchie appaiono meno profonde di quelle ottimali. Nei dati si evince infatti che l'*outdegree* medio è generalmente una quantità decrescente come funzione della distanza dalla radice, trend che viene interpretato dal modello come un regime non ottimale a causa di una profondità troppo bassa. Questo risultato è altresì confermato da generali istruzioni che in ambito informatico suggeriscono di limitare la complessità delle gerarchie in funzione di un più semplice utilizzo delle stesse.

La previsione dell'andamento dell'*outdegree* medio come funzione della distanza dalla radice nel modello Sharing Tree risulta una funzione ben sovrapponibile all'effettivo comportamento delle gerarchie nei linguaggi di programmazione.

Altre quantità nei dati sono risultate in pieno accordo con i modelli, come ad esempio la profondità delle strutture come funzione del numero di oggetti della gerarchia che si presenta in buona approssimazione logaritmica, comportamento previsto da entrambi i modelli.

L'ottimizzazione risultante dalla competizione tra il riuso e la genericità del codice sembra dunque giocare un ruolo importante nella costruzione delle gerarchie ereditarie nell'ambito della programmazione a oggetti.