Principles of Cryptocurrency Design

Appunti ed Esercizi

Francesco Pasquale

27 aprile 2026

Nella lezione di oggi abbiamo implementato la classe `Script` e un metodo che abbiamo utilizzato per fare il *parsing* degli script. Gli esercizi di questa nota si riferiscono al codice scritto in aula, che potete scaricare qui: `https://www.mat.uniroma2.it/~pasquale/dida/aa2526/pcd/pcd260427.zip`

Esercizio 1. Nel file `script.py` abbiamo implementato una classe *Script* e un metodo di classe `parse` che prende in input uno stream di byte consistente con la serializzazione degli script in Bitcoin e la lunghezza in byte dello script, e restituisce un oggetto *Script* con la sequenza di comandi codificati nello stream di byte.

Scrivere un metodo `serialize` per la classe *Script* che restituisca una sequenza di byte contenente la serializzazione dell'oggetto della classe.

Esercizio 2. Fare il parsing del seguente *locking script* `6e879169a87ca887`. Usando le descrizioni degli OP_CODES[1], determinare cosa dovrebbe contenere un *unlocking script* per ottenere uno script che restituisca TRUE.

Esercizio 3. Progettare un oggetto della classe Script che contenga due sequenze di byte, x e y, con y esattamente 32 byte, e tale che lo script restituisca TRUE se e solo se $\text{SHA256}(x) = y$.

Esercizio 4. Scrivere un interprete per un piccolo sottoinsieme del linguaggio *Script* (per esempio, considerando solo i comandi OP_DUP, OP_EQUAL, OP_VERIFY, OP_SWAP, OP_SHA256, OP_HASH256). L'interprete deve prendere in input l'istanza di un oggetto della classe Script, e restituire TRUE se lo *stack* al termine dell'esecuzione della lista dei comandi contiene un solo elemento che è 1, deve restituire FALSE in tutti gli altri casi.

[1] Le trovate, per esempio, qui https://wiki.bitcoinsv.io/index.php/Opcodes_used_in_Bitcoin_Script