

Relazione Homework Assignment 1

Fausto Francesco Frasca 559482

April 2021

1 Scelte progettuali

L'integrazione della stack inspection nell'interprete fornito è stata effettuata utilizzando una lista di liste come struttura di supporto, denominata `permissions_stack`. Un frame della `permissions_stack` è una lista che rappresenta l'intersezione dei permessi garantiti fino a quel momento. Tale scelta è stata effettuata per garantire un check più veloce dei permessi a run-time, andando a controllare esclusivamente la lista contenuta nel frame in cima alla `permissions_stack`.

Le operazioni di I/O e i relativi permessi implementati sono `read`, `write` ed `exec`.

Se la `permissions_stack` risulta essere vuota, in base alla politica dei privilegi minimi, verrà negata qualsiasi operazione di I/O. Ne consegue che tali operazioni possono essere effettuate solo all'interno di funzioni e che queste funzioni abbiano i privilegi necessari ad eseguirle.

Il primo frame della `permission_stack` conterrà la lista di tutti i diritti della funzione invocata.

2 Implementazione

2.1 Permessi ed Operazioni

È stato creato un tipo apposito per i permessi denominato *permission*, con valori *ReadPerm*, *WritePerm* e *ExecPerm*. Per quanto riguarda le operazioni, il tipo `expr` è stato esteso con le operazioni *Read*, *Write* ed *Exec*. Le operazioni di *Read* ed *Exec* prendono come parametro la stringa del file su cui eseguire l'operazione, mentre l'operazione di *Write* prende la coppia nome file, espressione da scrivere.

È stato modificato anche il costruttore `Fun`, adesso diventato una tripla. Sono rimasti invariati i precedenti parametri di input, ma è stato aggiunto come terzo parametro la lista dei permessi concessi alla funzione.

2.2 Funzioni ausiliarie per la gestione dei permessi

Oltre alle funzioni ausiliarie presenti nell'interprete originale, sono state aggiunte delle funzioni in supporto alla gestione dei permessi a run-time.

Le funzioni sono le seguenti:

- *lookup_permissions_stack* stack: ritorna il frame in cima allo stack se questo esiste, altrimenti la lista vuota.
- *permissions_frame* perm stack: ritorna l'intersezione tra perm e il frame in cima allo stack se quest'ultimo non è vuoto, altrimenti viene ritornato perm.
- *check_well_formed_permissions* perm: ritorna true se non ci sono permessi duplicati in perm, false altrimenti.
- *check_permissions* perm stack: ritorna true se perm è presente nella lista dei permessi contenuta nel frame in cima allo stack, false altrimenti.

2.3 Funzione di valutazione e valori a run-time

È stato opportunamente modificato il costruttore Closure. Closure è una quadrupla che come input aggiuntivo, oltre ai tre già presenti nell'interprete originale, prende la lista dei permessi specificati nella funzione associata.

Anche la funzione di valutazione *eval* è stata opportunamente modificata per integrare la stack inspection. La valutazione dell'espressione Fun, prima di ritornare la closure, effettua un controllo sui permessi, richiamando la funzione *check_well_formed_permissions*. Se il check ritorna false viene sollevata un'eccezione.

Per la valutazione dell'espressione Call è stata aggiunta la chiamata alla funzione *permissions_frame*, in modo da ottenere il frame da inserire in cima al *permissions_stack*. Infine sono state aggiunte le valutazioni delle operazioni di Read, Write ed Exec. Seguono tutte lo stesso pattern, ovvero viene effettuata una chiamata alla funzione *check_permissions* per accertarsi che si abbiano i permessi necessari per effettuare l'operazione. Se la check restituisce true viene ritornato un *Int 1*, altrimenti viene sollevata un'eccezione.