

PROGRAMMAZIONE II – A.A. 2015-16 – Secondo Progetto (15 dicembre) v.2

Il progetto ha l'obiettivo di applicare a casi specifici i concetti e le tecniche di programmazione esaminate durante la seconda parte del corso, e consiste nella progettazione e realizzazione di alcuni moduli software.

Descrizione: Progettazione e sviluppo di un interprete in OCaml

Si consideri un semplice linguaggio funzionale **miniPy** per la manipolazione di tuple di valori, la cui sintassi concreta è definita dalla grammatica riportata di seguito

(Identificatori)	Ide ::= <definizione standard>	
(Interi)	Int ::= <definizione standard>	
(Tuple)	T ::= () add(bVal, T)	
(Valori Base)	bVal ::= Int	(Interi)
	T	(Tuple)
	True False	(Valori Booleani)
(Valori Funzionali)	fVal ::= fun Ide -> E	(Funzioni non ricorsive)
Valori	Val :: bVal fVal	
(Espressioni)	E ::= Ide	(Identificatori)
	Val	(Valori)
	E and E E or E not E	(Espressioni Booleane)
	OP(E,E)	(Espressioni binarie interi)
	OP(E)	(Espressioni unarie interi)
	in(E,T)	(Test appartenenza di un valore a una tupla)
	isEmpty(T)	(Operatore su tuple – controlla se T è vuota)
	E[n:m]	(Slicing su tuple)
	E[n]	(Accesso a tuple)
	if E then E else E	(Condizionale)
	let Ide = E in E	(Blocco let)
	E(E)	(Applicazione Funzionale)
	for ide in T: E	(For loop)

Gli operatori hanno il significato intuitivo descritto tra parentesi. Meritano un commento specifico gli operatori di accesso, di slicing e il for loop.

Ad esempio sia t la tupla (23, True, (45, 7), False). Allora abbiamo

t[0] = 23, mentre t[3] = False e t[4] solleva una eccezione

t[-1] = False, mentre t[-3] = True e t[-5] solleva una eccezione

t[1:2] = (True, (45, 7)), mentre t[-1:-3] = (False, (45, 7), True) e t[-1:1] solleva una eccezione

for x in t: x+1 restituisce la tupla (24), mentre for x in t: fun y = y + x solleva una eccezione

1. Si definisca una sintassi astratta per il linguaggio, introducendo opportuni tipi di dati OCaml.
2. Si definisca in OCaml un interprete del linguaggio assumendo la regola di scoping statico.
3. Si verifichi la correttezza dell'interprete progettando ed eseguendo una quantità di casi di test sufficiente a testare tutti gli operatori.
4. Si descriva quali sono le modifiche da apportare all'interprete del linguaggio nel caso si voglia assumere una regola di scoping dinamico.

Modalità di consegna

- Il progetto deve essere svolto e discusso col docente individualmente. Il confronto con colleghi mirante a valutare soluzioni alternative durante la fase di progetto è incoraggiato.
- Il progetto deve essere costituito da
 - i file sorgente contenenti il codice sviluppato e le corrispondenti batterie di test, ove tutto il codice deve essere adeguatamente commentato;
 - una relazione di massimo una pagina che descrive le principali scelte progettuali ed eventuali istruzioni per eseguire il codice.
- La consegna va fatta inviando per email tutti i file in un archivio entro il 10 Gennaio 2016. Per il corso A, inviare l'email al Prof. Ferrari con oggetto "[PR2A] Consegna progetto 2". Per il corso B, inviare l'email al Prof. Gadducci con oggetto contenente la stringa "[PR2B] Consegna progetto 2".

Altre informazioni

- Per quanto riguarda il progetto, i docenti risponderanno solo a eventuali domande riguardanti l'interpretazione del testo, e non commenteranno soluzioni parziali prima della consegna.