Prova Scritta di Linguaggi di programmazione
Corso di Linguaggi di programmazione I
23/03/2005

Prolog

Scrivere in PROLOG puro un predicato `palindrome/1` che determina se una lista di elementi è palindroma. Quindi, per esempio, il goal `palindrome([a,b,b,a])` è vero.
Si scriva il programma con variabili anonime ove possibile.

```
palindrome(Xs) :- accumul(Xs,[]).

accumul([X|Xs],Ys) :- accumul(Xs,[X|Ys]).
accumul(Xs,Ys):-verify(Xs,Ys).
accumul([X|Xs],Ys):-verify(Xs,Ys).

verify([X|Xs],[X|Ys]) :- verify(Xs,Ys).
verify([],[]).
```

oppure la versione dichiarativa

```
palindrome(Xs) :- reverse(Xs,Xs).
```

dove reverse si può fare dichiarativo o efficiente

```
reverse(Xs) :- reverse(Xs,[]).

reverse([],Xs,Xs).
reverse([X|Xs],Ys,Zs):-reverse(Xs,[X|Ys],Zs).
```

Unificazione

Si dica se esiste un unificatore più generale di `p(X,f(T,a),h(X))` e `p(g(Z),T,Z)` e in caso affermativo lo si calcoli.

Non unifica.

Semantica

Si consideri il seguente frammento di programma espresso in un linguaggio imperativo C-like con scoping **dinamico**:

```
{       int x:= 3 ; int y:=2;
        proc P(ref int z; name int x)
        {
                z := 4+x;
                y := x+z+y;
        }
```

NOME E COGNOME:	Numero di Matricola:

```
        int y := 1;
    P(x, y+x);
    }
```

si fornisca la semantica denotazionale del frammento, mostrando l'ambiente e la memoria nel momento in cui termina la valutazione della chiamata di procedura P.

Implementazione

Con riferimento alla tecnica di implementazione con “shallow binding”, per l'esercizio precedente si mostri (schematicamente) con dei diagrammi la situazione sul vettore di sistema (per i valori) degli identificatori e sullo stack nascosto al momento in cui viene eseguita P.