

[Tablou de bord](#) / [Cursurile mele](#) / [2020-2021](#) / [Facultatea de Matematică și Informatică](#) / [Studii universitare de licență](#) / [Informatică](#) / [Limba română](#)
/ [Forma de învățământ - IF](#) / [Anul 2](#) / [Semestrul II](#) / [Fundamentele limbajelor de programare - Seria 23](#) / General / [Examen \(ora 10:00\)](#)

Început pe	vineri, 25 iunie 2021, 09:59
Status	Finalizat
Completat pe	vineri, 25 iunie 2021, 11:51
Timp de parcursere test	1 oră 52 min
Notă obținută	Nu este încă notat

1 întrebare

Complet

Marcat din 1,00

Să se descrie detaliat (regula aplicată, unificare cap, obținerea noii ținte,...) execuția țintei indicate pentru programul Prolog de mai jos:

Program Prolog:

$p(c,b).$

$p(a,b).$

$r(b,c).$

$r(b,a).$

$r(X,Z) :- r(Y,X), r(Y,Z).$

$r(X,Y) :- p(Y,X).$

Ținta

?- $r(a,Y)$

Pentru a rezolva $r(a, Y)$ putem aplica regulile (5) si (6)

Aplicam regula (5):

- Redenumim regula (5) in $r(X1, Z1) :- r(Y1, X1), r(Y1, Z1)$
- Unificam $r(a, Y)$ cu $r(X1, Z1)$ - obținem $\theta_1 = X1 = a, Z1 = Y$
- Inlocuim $r(a, Y)$ cu $r(Y1, X1), r(Y1, Z1)$ si aplicam substitutia θ_1
- Obținem noua tinta $r(Y1, a), r(Y1, Y)$
- Trebuie sa rezolvam $r(Y1, a)$. Putem aplica regulile (3), (4)
 - Aplicam regula (3). Nu avem ce redenumi
 - Unificam $r(Y1, a)$ cu $r(b, c)$ - unificare esuata, backtrack.
 - Aplicam regula (4). Nu avem ce redenumi.
 - Unificam $r(Y1, a)$ cu $r(b, a)$ - obținem $\theta_2 = Y1 = b$
 - Inlocuim $r(Y1, a)$ cu \emptyset si aplicam substitutia θ_2
 - Obținem noua tinta $r(b, Y)$
 - Putem aplica regulile (3) si (4).
 - Aplicam regula (3). Nu avem ce redenumi
 - Unificam $r(b, Y)$ cu $r(b, c)$ - obținem $\theta_3 = Y = c$
 - Inlocuim cu \emptyset si aplicam substitutia θ_3
 - Obținem tinta vida. Avem solutia $\theta = Y = c$
 - Aplicam regula (4). Nu avem ce redenumi
 - Unificam $r(b, Y)$ cu $r(b, a)$ - obținem $\theta_4 = Y = a$
 - Inlocuim cu \emptyset si aplicam substitutia θ_4
 - Obținem tinta vida. Avem solutiile $\theta = Y = a$.

Aplicam regula (6):

- Redenumim regula (6) in $r(X2, Y2) :- p(Y2, X2)$
- Unificam $r(a, Y)$ cu $r(X2, Y2)$ - obținem $\theta_1 = X2 = a, Y2 = Y$
- Inlocuim $r(a, Y)$ cu $p(Y2, X2)$ si aplicam substitutia θ_1
- Obținem noua tinta $p(Y, a)$. Putem aplica regulile (1) si (2)
 - Aplicam regula (1). Nu avem ce redenumi
 - Unificam $p(Y, a)$ cu $p(c, b)$ - unificare esuata, backtrack
 - Aplicam regula (2). Nu avem ce redenumi
 - Unificam $p(Y, a)$ cu $p(a, b)$ - unificare esuata, backtrack

Concluzie: $Y = \{c, a\}$



ex1.2.pdf

2 întrebare

Complet

Marcat din 1,00

În acest exercițiu: a,b,c sunt simboluri de constante; x,y,z,u,v,w sunt variabile; f,g,h,p sunt simboluri de funcții, aritatea fiind dedusa din context.

Aplicând algoritmul de unificare din curs, verificați dacă exista un unificator pentru următorii termeni (detaliați fiecare pas din algoritm):

$p(f(x,y), f(h(x),x), z)$ și $p(z,z,f(x,b))$

Initial	$p(f(x, y), f(h(x), x), z) = p(z, z, f(x, b))$	
Descompune	$f(x,y) = z, f(h(x), x) = z, z = f(x, b)$	
Rezolva	$z = f(x, y)$	$f(h(x), x) = f(x, y), f(x, y) = f(x, b)$
Descompune	$z = f(x, y)$	$h(x) = x, x = y, x = x, y = b$

In ecuatia $h(x) = x$, variabila x apare in termenul general $h(x)$.

Nu exista unificator.

3 întrebare

Complet

Marcat din 1,00

Pentru acest exercițiu folosim limbajul IMP definit în curs.

Folosind regulile semanticii operaționale "small step" definite în curs justificați pașii de tranziție ceruți de exercițiu, indicând la fiecare pas regula folosită și termenul prelucrat de regula.

Atenție: ``mvid`` este starea memoriei vida, iar o stare a memoriei va fi reprezentată prin $\{x \rightarrow vx, y \rightarrow vy, \dots\}$ unde x,y,... sunt variabile, iar vx,yy,... sunt valori.

Executați 5 pași de execuție din configurația:

```
< while (i*i <= 25, i = i + 1), {i->1} >
```

Pas 0: $\langle \text{while } (i*i \leq 25, i = i + 1), \{i \rightarrow 1\} \rangle \rightarrow (\text{While})$

Pas 1: $\langle \text{if } (i*i \leq 25, i = i + 1; \text{while } (i*i \leq 25, i = i + 1), \text{skip}), \{i \rightarrow 1\} \rangle \rightarrow (\text{Id})$

Pas 2: $\langle \text{if } (1*i \leq 25, i = i + 1; \text{while } (i*i \leq 25, i = i + 1), \text{skip}), \{i \rightarrow 1\} \rangle \rightarrow (\text{Id})$

Pas 3: $\langle \text{if } (1*i \leq 25, i = i + 1; \text{while } (i*i \leq 25, i = i + 1), \text{skip}), \{i \rightarrow 1\} \rangle \rightarrow (\text{Calc})$

Pas 4: $\langle \text{if } (1 \leq 25, i = i + 1; \text{while } (i*i \leq 25, i = i + 1), \text{skip}), \{i \rightarrow 1\} \rangle \rightarrow (\text{Leq-True})$

Pas 5: $\langle \text{if } (\text{true}, i = i + 1; \text{while } (i*i \leq 25, i = i + 1), \text{skip}), \{i \rightarrow 1\} \rangle$

4 întrebare

Complet

Marcat din 1,00

Consideram sistemul de inferenta a tipurilor pentru limbajul LAMBDA prezentat la curs (accesibil in ``lambdtypeoriginal.pl`` la link-ul: <http://tinyurl.com/lambdtype>) si vrem sa adaugam la limbaj date de tipul ``either(a,b)`` unde ``a`` si ``b`` sunt tipuri arbitrare.

Pentru aceasta avem nevoie e o modalitate de a construi noile date și de o modalitate pentru a le "deconstrui". Ca să realizăm acest lucru, adăugăm la LAMBDA următoarele functii:

- ``left`` care primește un argument de un tip arbitrar ``a`` si intoarce o valoare de tip ``either(a,b)``,
- ``right`` care primește un argument de un tip arbitrar ``b`` si intoarce o valoare de tip ``either(a,b)``,
- ``either`` care intoarce un rezultat de tipul ``rez`` primind ca argumente o valoare de tip ``either(a,b)``, o functie care are prelucreaza o valoare de tip ``a`` si intoarce un rezultat de tip ``rez`` si o functie care are prelucreaza o valoare de tip ``b`` si intoarce un rezultat de tip ``rez``.

Extindeti sistemul de inferenta a tipurilor astfel adaugand reguli pentru functiile ``left``, ``right`` si ``either``.

Precizati in caseta de raspuns numai regulile noi.

Testare:

In fisierul ``lambdtypeoriginal.pl`` inlocuiti definitia predicatului ``add_context`` cu cea de mai jos si testati daca programul determina corect tipurile functiilor definite de aceasta:

```
add_context(X,
  let(id, x -> x,
    let(const, x -> y -> x,
      let(is_left, e -> either $ e $ (a -> true) $ (b -> false),
        let(is_right, e -> either $ e $ (a -> false) $ (b -> true),
          let(from_left, a -> e -> either $ e $ id $ (const $ a),
            let(from_right, b -> e -> either $ e $ (const $ b) $ id,
              X
            ))))))).
% run(is_left).
% run(is_right).
% run(from_left).
% run(from_right).
```

```
type(_, left, _ -> either).
type(_, right, _ -> either).
type(_, either, either -> (_ -> R) -> (_ -> R) -> R).
```

5 întrebare

Complet

Marcat din 1,00

Pentru lambda-expresia de mai jos, indicați pentru fiecare apariție a unei variabile dacă este variabilă de legătură, variabilă liberă sau variabilă legată. Pentru fiecare variabilă legată indicați apariția variabilei de legătură de care aceasta este legată.

Puteți redenumi/sublinia, colora aparițiile variabilelor, dar trebuie să includeți o explicație a modului în care ați făcut acest lucru.

Se aplică convențiile sintactice precizate în curs.

$(\lambda r.(\lambda e.(\lambda t.r \ e \ c) (((\lambda c.c \ c)(r + e)) + r))) (\lambda r.r + c)$

$(\lambda r^1.(\lambda e^2.(\lambda t^3.r^4 \ e^5 \ c^6) (((\lambda c^7.c^8 \ c^9)(r^{10} + e^{11})) + r^{12}))) (\lambda r^{13}.r^{14} + c^{15})$

- r^1 variabila de legatura
- e^2 variabila de legatura
- t^3 variabila de legatura
- r^4 este legat de r^1
- e^5 este legat de e^2
- c^6 este libera
- c^7 variabila de legatura
- c^8 este legat de c^7
- c^9 este legat de c^7
- r^{10} este legat de r^1
- e^{11} este legat de e^2
- r^{12} este legat de r^1
- r^{13} variabila de legatura
- r^{14} este legat de r^{13}
- c^{15} este libera

◀ Repetare parțial (ora 8:00)

Sari la...

