

Zadání: Podrobně popište vyhodnocení následujícího s-výrazu. Nakreslete hierarchii prostředí, která při tomto vyhodnocení vznikne. Prostředí číslujte podle pořadí, ve kterém vznikají.

```
(define pokus (lambda (x) (+ x 1)))  
  
(- (pokus 10) 7)
```

Poznámka: Tento příklad je přítomen hlavně proto, aby bylo vidět jak jsou zapisována řešení. Šipky zde vedou od vyhodnocovaného výrazu k jeho vyhodnocení přes vše, co se během tohoto vyhodnocení děje. Barvy šipek nemají význam, pro šipky jsou používány různé barvy pro snadnější ztotožňování začátků a konců šipek v případě, že vedou na novou stránku. Vznik a modifikace prostředí jsou zvýrazněny červenou barvou.

Řešení:

vyhodnocujeme `(define pokus (lambda (x) (+ x 1)))` v prostředí \mathcal{P}_g .
je to seznam; nejprve vyhodnotíme první prvek seznamu.

vyhodnocujeme `define` v prostředí \mathcal{P}_g .

Je to symbol.

Nalezena vazba `define` \rightarrow speciální forma *define* v prostředí \mathcal{P}_g .

\rightarrow výsledek: *speciální forma define*

první prvek se vyhodnotil na speciální formu.

speciální forma define je aplikována na argumenty `pokus` a `(lambda (x) (+ x 1))`.

vyhodnocujeme `(lambda (x) (+ x 1))` v prostředí \mathcal{P}_g .

je to seznam; nejprve vyhodnotíme první prvek seznamu.

vyhodnocujeme `lambda` v prostředí \mathcal{P}_g .

Je to symbol.

Nalezena vazba `lambda` \rightarrow speciální forma *lambda* v prostředí \mathcal{P}_g .

\rightarrow výsledek: *speciální forma lambda*

první prvek se vyhodnotil na speciální formu.

speciální forma lambda je aplikována na argumenty `(x)` a `(+ x 1)`;

\rightarrow výsledek: $\langle (x), (+ x 1), \mathcal{P}_g \rangle$

do prostředí \mathcal{P}_g . je přidána vazba symbolu `pokus` $\rightarrow \langle (x), (+ x 1), \mathcal{P}_g \rangle$

\rightarrow výsledek: *nedefinovaná hodnota*

vyhodnocujeme `(- (pokus 10) 7)` v prostředí \mathcal{P}_g .

je to seznam; nejprve vyhodnotíme první prvek seznamu.

vyhodnocujeme `-` v prostředí \mathcal{P}_g .

Je to symbol.

Nalezena vazba `-` \rightarrow primitivní procedura *odčítání čísel* v prostředí \mathcal{P}_g .

\rightarrow výsledek: *primitivní procedura odčítání čísel*

je to procedura, vyhodnotíme zbytek seznamu.

vyhodnocujeme `(pokus 10)` v prostředí \mathcal{P}_g .

je to seznam; nejprve vyhodnotíme první prvek seznamu.

vyhodnocujeme **pokus** v prostředí \mathcal{P}_g .
 Je to symbol.
 Nalezena vazba **pokus** $\rightarrow \langle (x), (+ x 1), \mathcal{P}_g \rangle$ v prostředí \mathcal{P}_g .
 \rightarrow výsledek: $\langle (x), (+ x 1), \mathcal{P}_g \rangle$

je to procedura, vyhodnotíme zbytek seznamu.
 vyhodnocujeme **10** v prostředí \mathcal{P}_g .
 Je to číslo, vyhodnotí se samo na sebe.
 \rightarrow výsledek: 10

uživ. def. procedura $\langle (x), (+ x 1), \mathcal{P}_g \rangle$ je aplikována na argument 10;
Vytvořeno nové prostředí \mathcal{P}_1 ; jeho předek je \mathcal{P}_g .
Do prostředí \mathcal{P}_1 je zavedena vazba $x \rightarrow 10$;

vyhodnocujeme **(+ x 1)** v prostředí \mathcal{P}_1
 je to seznam; nejprve vyhodnotíme první prvek seznamu.
 vyhodnocujeme **+** v prostředí \mathcal{P}_1
 Je to symbol.
 Nalezena vazba **+** \rightarrow *primitivní procedura sčítání čísel* v prostředí \mathcal{P}_g .
 \rightarrow výsledek: *primitivní procedura sčítání čísel*

je to procedura, vyhodnotíme zbytek seznamu.
 vyhodnocujeme **x** v prostředí \mathcal{P}_1
 Je to symbol.
 Nalezena vazba **x** $\rightarrow 10$ v prostředí \mathcal{P}_1 .
 \rightarrow výsledek: 10

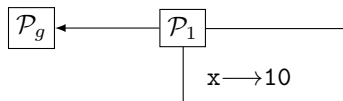
vyhodnocujeme **1** v prostředí \mathcal{P}_1 .
 Je to číslo, vyhodnotí se samo na sebe.
 \rightarrow výsledek: 1

primitivní procedura sčítání čísel je aplikována na argumenty 10 a 1;
 \rightarrow výsledek: 11

\rightarrow výsledek: 11

vyhodnocujeme **7** v prostředí \mathcal{P}_g .
 Je to číslo, vyhodnotí se samo na sebe.
 \rightarrow výsledek: 7

primitivní procedura odčítání čísel je aplikována na argumenty 11 a 7;
 \rightarrow výsledek: 4



Poznámka: Další řešení už budou mírně zkrácena.

Zadání: Podrobně popište vyhodnocení následujícího s-výrazu. Nakreslete hierarchii prostředí, která při tomto vyhodnocení vznikne. Prostředí číslujte podle pořadí, ve kterém vznikají.

```
(let* ((x (lambda (x . y) (listt x y)))
      (y (lambda (y) (x y y))))
      (map y (quote (1 2 3))))
```

Řešení:

vyhodnocujeme

```
(let* ((x (lambda (x . y) (list x y)))
      (y (lambda (y) (x y y))))
      (map y (quote (1 2 3))))
```

v prostředí \mathcal{P}_g .

$\text{let*} \longrightarrow$ speciální forma let* (vazba v prostředí \mathcal{P}_g)
sémantika odpovídá vyhodnocení výrazu:

```
((lambda (x)
  (let* ((y (lambda (y) (x y y)))
        (map y (quote (1 2 3)))))
  (lambda (x . y) (list x y)))
```

vyhodnocujeme tedy

```
((lambda (x)
  (let* ((y (lambda (y) (x y y)))
        (map y (quote (1 2 3)))))
  (lambda (x . y) (list x y)))
```

v prostředí \mathcal{P}_g .

vyhodnocujeme

```
(lambda (x)
  (let* ((y (lambda (y) (x y y)))
        (map y (quote (1 2 3)))))
```

v prostředí \mathcal{P}_g .

$\text{lambda} \longrightarrow$ spec. forma lambda (vazba v prostředí \mathcal{P}_g)

spec. forma lambda je aplikována na argumenty

(x) a $(\text{let*} ((y \dots)) (\text{map y (quote (1 2 3))}))$;

\longrightarrow výsledek: $\langle (x), (\text{let*} ((y \dots)) (\text{map y (quote (1 2 3))})) \rangle, \mathcal{P}_g$

vyhodnocujeme $(\text{lambda (x . y) (list x y)})$ v prostředí \mathcal{P}_g

$\text{lambda} \longrightarrow$ spec. forma lambda (vazba v prostředí \mathcal{P}_g)

spec. forma lambda je aplikována na argument $\langle x . y \rangle$ (`list x y`);
→ výsledek: $\langle (x . y), (\text{list } x \ y), \mathcal{P}_g \rangle$

uživ. def. proc. $\langle (x), (\text{let* } ((y \ \dots)) (\text{map } y (\text{quote } (1 \ 2 \ 3)))) \rangle, \mathcal{P}_g$
je aplikována na argument $\langle (x . y), (\text{list } x \ y), \mathcal{P}_g \rangle$;

Vytvořeno nové prostředí \mathcal{P}_1 ; jeho předek je \mathcal{P}_g

Do prostředí \mathcal{P}_1 je zavedena vazba: $x \rightarrow \langle (x . y), (\text{list } x \ y), \mathcal{P}_g \rangle$;

vyhodnocujeme

```
(let* ((y (lambda (y) (x y y)))  
      (map y (quote (1 2 3))))
```

v prostředí \mathcal{P}_1 .

let* → *speciální forma let** (vazba v prostředí \mathcal{P}_g)

sémantika odpovídá vyhodnocení výrazu:

```
((lambda (y)  
  (let* () (map y (quote (1 2 3)))))  
 (lambda (y) (x y y)))
```

vyhodnocujeme tedy

```
((lambda (y)  
  (let* () (map y (quote (1 2 3)))))  
 (lambda (y) (x y y)))
```

v prostředí \mathcal{P}_1 .

vyhodnocujeme

```
(lambda (y) (map y (quote (1 2 3))))
```

v prostředí \mathcal{P}_1 .

lambda → *spec. forma lambda* (vazba v prostředí \mathcal{P}_g)

spec. forma lambda je aplikována na argumenty:

$\langle y \rangle$ a $\langle \text{map } y (\text{quote } (1 \ 2 \ 3)) \rangle$;

→ výsledek: $\langle (y), (\text{map } y (\text{quote } (1 \ 2 \ 3))), \mathcal{P}_1 \rangle$

vyhodnocujeme $\langle \text{lambda } (y) (x \ y \ y) \rangle$ v prostředí \mathcal{P}_1 .

lambda → *spec. forma lambda* (vazba v prostředí \mathcal{P}_g)

spec. forma lambda je aplikována na argument $\langle (y) (x \ y \ y) \rangle$;

→ výsledek: $\langle (y), (x \ y \ y), \mathcal{P}_1 \rangle$

uživ. def. procedura $\langle (y), (\text{map } y (\text{quote } (1 \ 2 \ 3))), \mathcal{P}_1 \rangle$ je aplikována
na argument $\langle (y), (x \ y \ y), \mathcal{P}_1 \rangle$;

Vytvořeno nové prostředí \mathcal{P}_2 ; jeho předek je \mathcal{P}_1

Do prostředí \mathcal{P}_2 jsou zavedeny vazby: $y \rightarrow \langle (y), (x \ y \ y), \mathcal{P}_1 \rangle$;

vyhodnocujeme $\langle \text{map } y (\text{quote } (1 \ 2 \ 3)) \rangle$ v prostředí \mathcal{P}_2 .

map → *primitivní procedura map* (vazba v prostředí \mathcal{P}_g)

y → $\langle (y), (x \ y \ y), \mathcal{P}_1 \rangle$ (vazba v prostředí \mathcal{P}_2)

vyhodnocujeme `(quote (1 2 3))` v prostředí \mathcal{P}_2 .
`quote` \longrightarrow *spec. forma quote* (vazba v prostředí \mathcal{P}_g)
spec. forma quote je aplikována na argument `(1 2 3)`;
 \longrightarrow výsledek: `(1 2 3)`

primitivní procedura map je aplikována na argumenty $\langle (y), (x\ y\ y), \mathcal{P}_1 \rangle$
a `(1 2 3)`;

uživ. def. procedura $\langle (y), (x\ y\ y), \mathcal{P}_1 \rangle$ je aplikována na argument 1;

Vytvořeno nové prostředí \mathcal{P}_3 ; jeho předek je \mathcal{P}_1

Do prostředí \mathcal{P}_3 je zavedena vazba: $y \longrightarrow 1$;

vyhodnocujeme `(x y y)` v prostředí \mathcal{P}_3 .
`x` \longrightarrow $\langle (x\ .\ y), (\text{list } x\ y), \mathcal{P}_g \rangle$ (vazba v prostředí \mathcal{P}_1)
`y` \longrightarrow 1 (vazba v prostředí \mathcal{P}_3)
`y` \longrightarrow 1 (vazba v prostředí \mathcal{P}_3)
uživ. def. procedura $\langle (x\ .\ y), (\text{list } x\ y), \mathcal{P}_g \rangle$ je aplikována na argu-
menty 1 a 1;

Vytvořeno nové prostředí \mathcal{P}_4 ; jeho předek je \mathcal{P}_g

Do prostředí \mathcal{P}_4 jsou zavedeny vazby: $x \longrightarrow 1$; $y \longrightarrow (1)$;

vyhodnocujeme `(list x y)` v prostředí \mathcal{P}_4 .
`list` \longrightarrow *primitivní procedura list* (vazba v prostředí \mathcal{P}_g)
`x` \longrightarrow 1 (vazba v prostředí \mathcal{P}_4)
`y` \longrightarrow (1) (vazba v prostředí \mathcal{P}_4)
primitivní procedura list je aplikována na argumenty 1 a (1);
 \longrightarrow výsledek: `(1 (1))`
 \longrightarrow výsledek: `(1 (1))`

uživ. def. procedura $\langle (y), (x\ y\ y), \mathcal{P}_1 \rangle$ je aplikována na argument 2;

Vytvořeno nové prostředí \mathcal{P}_5 ; jeho předek je \mathcal{P}_1

Do prostředí \mathcal{P}_5 je zavedena vazba: $y \longrightarrow 2$;

vyhodnocujeme `(x y y)` v prostředí \mathcal{P}_5 .
`x` \longrightarrow $\langle (x\ .\ y), (\text{list } x\ y), \mathcal{P}_g \rangle$ (vazba v prostředí \mathcal{P}_1)
`y` \longrightarrow 2 (vazba v prostředí \mathcal{P}_5)
`y` \longrightarrow 2 (vazba v prostředí \mathcal{P}_5)
uživ. def. procedura $\langle (x\ .\ y), (\text{list } x\ y), \mathcal{P}_g \rangle$ je aplikována na argu-
menty 2 a 2;

Vytvořeno nové prostředí \mathcal{P}_6 ; jeho předek je \mathcal{P}_g

Do prostředí \mathcal{P}_6 jsou zavedeny vazby: $x \longrightarrow 2$; $y \longrightarrow (2)$;

vyhodnocujeme `(list x y)` v prostředí \mathcal{P}_6 .
`list` \longrightarrow *primitivní procedura list* (vazba v prostředí \mathcal{P}_g)
`x` \longrightarrow 2 (vazba v prostředí \mathcal{P}_6)
`y` \longrightarrow (2) (vazba v prostředí \mathcal{P}_6)
primitivní procedura list je aplikována na argumenty 2 a (2);
 \longrightarrow výsledek: `(2 (2))`
 \longrightarrow výsledek: `(2 (2))`

uživ. def. procedura $\langle (y), (x\ y\ y), \mathcal{P}_1 \rangle$ je aplikována na argument 3;

Vytvořeno nové prostředí \mathcal{P}_7 ; jeho předek je \mathcal{P}_1
 Do prostředí \mathcal{P}_7 jsou zavedeny vazby: $y \rightarrow 3$;

vyhodnocujeme $(x\ y\ y)$ v prostředí \mathcal{P}_7 .

$x \rightarrow \langle (x\ .\ y), (\text{list } x\ y), \mathcal{P}_g \rangle$ (vazba v prostředí \mathcal{P}_1)
 $y \rightarrow 3$ (vazba v prostředí \mathcal{P}_7)
 $y \rightarrow 3$ (vazba v prostředí \mathcal{P}_7)

uživ. def. procedura $\langle (x\ .\ y), (\text{list } x\ y), \mathcal{P}_g \rangle$ je aplikována na argumenty 3 a 3;

Vytvořeno nové prostředí \mathcal{P}_8 ; jeho předek je \mathcal{P}_g
 Do prostředí \mathcal{P}_8 jsou zavedeny vazby: $x \rightarrow 3$; $y \rightarrow (3)$;

vyhodnocujeme $(\text{list } x\ y)$ v prostředí \mathcal{P}_8 .

$\text{list} \rightarrow \text{primitivní procedura list}$ (vazba v prostředí \mathcal{P}_g)
 $x \rightarrow 3$ (vazba v prostředí \mathcal{P}_8)
 $y \rightarrow (3)$ (vazba v prostředí \mathcal{P}_8)

primitivní procedura list je aplikována na argumenty 3 a (3);

\rightarrow výsledek: (3 (3))

\rightarrow výsledek: (3 (3))

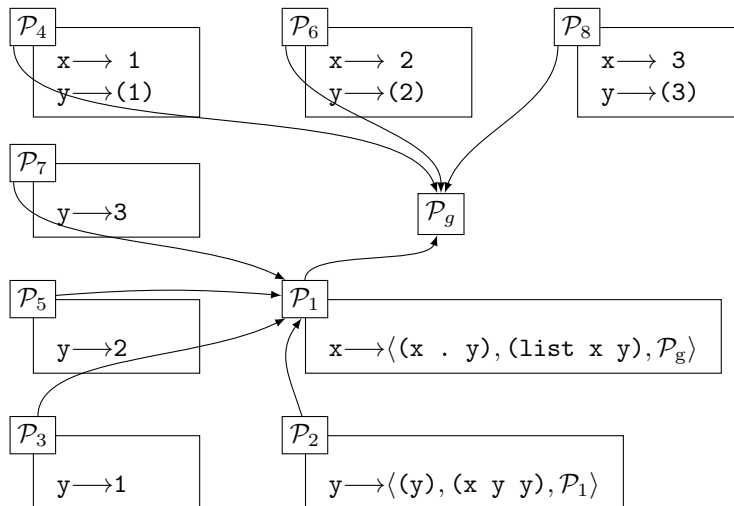
\rightarrow výsledek: ((1 (1)) (2 (2)) (3 (3)))

\rightarrow výsledek: ((1 (1)) (2 (2)) (3 (3)))

\rightarrow výsledek: ((1 (1)) (2 (2)) (3 (3)))

\rightarrow výsledek: ((1 (1)) (2 (2)) (3 (3)))

\rightarrow výsledek: ((1 (1)) (2 (2)) (3 (3)))



Poznámka: Protože není specifikováno, v jakém pořadí *map* aplikuje proceduru, mohou prostředí \mathcal{P}_3 , \mathcal{P}_5 a \mathcal{P}_7 (a tedy i \mathcal{P}_4 , \mathcal{P}_6 a \mathcal{P}_8) vznikat v jiném pořadí, než říká jejich číslování.

Zadání: Podrobně popište vyhodnocení následujícího s-výrazu. Nakreslete hierarchii prostředí, která při tomto vyhodnocení vznikne. Prostředí číslujte podle pořadí, ve kterém vznikají.

```
((lambda (x . y)
  (let ((x (list x y)) (- (lambda (x y) (cons x y))))
    ((lambda (z)
      (list (z x (car y))
            (z 10 x))) -))) ' x 'y)
```

Řešení:

vyhodnocujeme

```
((lambda (x . y)
  (let ((x (list x y)) (- (lambda (x y) (cons x y))))
    ((lambda (z)
      (list (z x (car y))
            (z 10 x))) -))) ' x 'y)
```

v prostředí \mathcal{P}_g

vyhodnocujeme

```
(lambda (x . y)
  (let ((x (list x y)) (- (lambda (x y) (cons x y))))
    ((lambda (z)
      (list (z x (car y))
            (z 10 x))) -)))
```

v prostředí \mathcal{P}_g

lambda \longrightarrow *spec. forma lambda* (vazba v prostředí \mathcal{P}_g)

spec. forma lambda je aplikována na argumenty $(x . y)$ a $(let \dots)$;

\longrightarrow výsledek: $\langle (x . y), (let \dots), \mathcal{P}_g \rangle$

vyhodnocujeme **(quote x)** v prostředí \mathcal{P}_g

quote \longrightarrow *spec. forma quote* (vazba v prostředí \mathcal{P}_g)

spec. forma quote je aplikována na argumenty (x) ;

\longrightarrow výsledek: x

vyhodnocujeme **(quote y)** v prostředí \mathcal{P}_g

quote \longrightarrow *spec. forma quote* (vazba v prostředí \mathcal{P}_g)

spec. forma quote je aplikována na argumenty (y) ;

\longrightarrow výsledek: y

uživ. def. procedura $\langle (x . y), (let \dots), \mathcal{P}_g \rangle$ je aplikována na argumenty x a y ;

Vytvořeno nové prostředí \mathcal{P}_1 ; jeho předek je \mathcal{P}_g

Do prostředí \mathcal{P}_1 jsou zavedeny vazby: $x \longrightarrow x$; $y \longrightarrow (y)$;

vyhodnocujeme

```
(let ((x (list x y))
      (- (lambda (x y) (cons x y))))
      ((lambda (z)
         (list (z x (car y))
               (z 10 x))) -))
```

v prostředí \mathcal{P}_1 .

let \longrightarrow *speciální forma* (vazba v prostředí \mathcal{P}_g)
sémantika odpovídá vyhodnocení výrazu:

```
((lambda (x -)
  ((lambda (z)
    (list (z x (car y))
          (z 10 x))) -))
 (list x y) (lambda (x y) (cons x y)))
```

vyhodnocujeme tedy

```
((lambda (x -)
  ((lambda (z)
    (list (z x (car y))
          (z 10 x))) -))
 (list x y) (lambda (x y) (cons x y)))
```

v prostředí \mathcal{P}_1 .

vyhodnocujeme

```
((lambda (x -)
  ((lambda (z)
    (list (z x (car y))
          (z 10 x))) -))
```

v prostředí \mathcal{P}_1 .

lambda \longrightarrow *spec. forma lambda* (vazba v prostředí \mathcal{P}_g)
spec. forma lambda je aplikována na argumenty $(x -)$ a $((lambda (z) \dots) -)$;

\longrightarrow výsledek: $\langle (x -), ((lambda (z) \dots) -), \mathcal{P}_1 \rangle$

vyhodnocujeme **(list x y)** v prostředí \mathcal{P}_1 .

list \longrightarrow *primitivní procedura list* (vazba v prostředí \mathcal{P}_g)

x \longrightarrow **x** (vazba v prostředí \mathcal{P}_1)

y \longrightarrow **(y)** (vazba v prostředí \mathcal{P}_1)

primitivní procedura list je aplikována na argumenty **x** a **(y)**;

\longrightarrow výsledek: **(x (y))**

vyhodnocujeme **(lambda (x y) (cons x y))** v prostředí \mathcal{P}_1 .

lambda \longrightarrow *spec. forma lambda* (vazba v prostředí \mathcal{P}_g)

spec. forma lambda je aplikována na argumenty $(x\ y)$ a $(\text{cons } x\ y)$;
→ výsledek: $\langle (x\ y), (\text{cons } x\ y), \mathcal{P}_1 \rangle$

uživ. def. procedura $\langle (x\ -), ((\text{lambda } (z) \dots) -), \mathcal{P}_1 \rangle$ je aplikována na argumenty $(x\ (y))$ a $\langle (x\ y), (\text{cons } x\ y), \mathcal{P}_1 \rangle$;

Vytvořeno nové prostředí \mathcal{P}_2 ; jeho předek je \mathcal{P}_1 .

Do prostředí \mathcal{P}_2 jsou zavedeny vazby: $x \rightarrow (x\ (y))$; $- \rightarrow \langle (x\ y), (\text{cons } x\ y), \mathcal{P}_1 \rangle$;

vyhodnocujeme

$((\text{lambda } (z) (\text{list } (z\ x\ (\text{car } y)) (z\ 10\ x))) -)$

v prostředí \mathcal{P}_2 .

vyhodnocujeme

$(\text{lambda } (z) (\text{list } (z\ x\ (\text{car } y)) (z\ 10\ x)))$ v prostředí \mathcal{P}_2 .

$\text{lambda} \rightarrow$ *spec. forma lambda* (vazba v prostředí \mathcal{P}_g)

spec. forma lambda je aplikována na argumenty (z) a $(\text{list } (z\ x\ (\text{car } y)) (z\ 10\ x))$;

→ výsledek: $\langle (z), (\text{list } (z\ x\ (\text{car } y)) (z\ 10\ x)), \mathcal{P}_2 \rangle$

$- \rightarrow \langle (x\ y), (\text{cons } x\ y), \mathcal{P}_1 \rangle$ (vazba v prostředí \mathcal{P}_2)

uživ. def. procedura $\langle (z), (\text{list } (z\ x\ (\text{car } y)) (z\ 10\ x)), \mathcal{P}_2 \rangle$ je aplikována na argumenty $\langle \langle (x\ y), (\text{cons } x\ y), \mathcal{P}_1 \rangle \rangle$;

Vytvořeno nové prostředí \mathcal{P}_3 ; jeho předek je \mathcal{P}_2 .

Do prostředí \mathcal{P}_3 jsou zavedeny vazby: $z \rightarrow \langle (x\ y), (\text{cons } x\ y), \mathcal{P}_1 \rangle$;

vyhodnocujeme $(\text{list } (z\ x\ (\text{car } y)) (z\ 10\ x))$ v prostředí \mathcal{P}_3

$\text{list} \rightarrow$ *primitivní procedura list* (vazba v prostředí \mathcal{P}_g)

vyhodnocujeme $(z\ x\ (\text{car } y))$ v prostředí \mathcal{P}_3 .

$z \rightarrow \langle (x\ y), (\text{cons } x\ y), \mathcal{P}_1 \rangle$ (vazba v prostředí \mathcal{P}_3)

$x \rightarrow (x\ (y))$ (vazba v prostředí \mathcal{P}_2)

vyhodnocujeme $(\text{car } y)$ v prostředí \mathcal{P}_3 .

$\text{car} \rightarrow$ *primitivní procedura car* (vazba v prostředí \mathcal{P}_g)

$y \rightarrow (y)$ (vazba v prostředí \mathcal{P}_1)

primitivní procedura car je aplikována na argument (y) ;

→ výsledek: y

uživ. def. procedura $\langle (x\ y), (\text{cons } x\ y), \mathcal{P}_1 \rangle$ je aplikována na argument $(x\ (y))\ y$;

Vytvořeno nové prostředí \mathcal{P}_4 ; jeho předek je \mathcal{P}_1 .

Do prostředí \mathcal{P}_4 jsou zavedeny vazby: $x \rightarrow (x\ (y))$; $y \rightarrow y$;

vyhodnocujeme $(\text{cons } x\ y)$ v prostředí \mathcal{P}_4 .

$\text{cons} \rightarrow$ *primitivní procedura cons* (vazba v prostředí \mathcal{P}_g)

$x \rightarrow (x\ (y))$ (vazba v prostředí \mathcal{P}_4)

$y \rightarrow y$ (vazba v prostředí \mathcal{P}_4)

primitivní procedura cons je aplikována na argumenty $((x\ (y))\ y)$;

→ výsledek: $((x\ (y))\ .\ y)$

→ výsledek: $((x\ (y))\ .\ y)$

vyhodnocujeme $(z\ 10\ x)$ v prostředí \mathcal{P}_3 .

$10 \longrightarrow 10$
 $x \longrightarrow (x (y))$ (vazba v prostředí \mathcal{P}_2)
 uživ. def. procedura $\langle (x y), (\text{cons } x y), \mathcal{P}_1 \rangle$ je aplikována na argumenty $(10 (x (y)))$;
Vytvořeno nové prostředí \mathcal{P}_5 ; jeho předek je \mathcal{P}_1
 Do prostředí \mathcal{P}_5 jsou zavedeny vazby: $x \longrightarrow 10$; $y \longrightarrow (x (y))$;
 vyhodnocujeme $(\text{cons } x y)$ v prostředí \mathcal{P}_5 .
 $\text{cons} \longrightarrow$ primitivní procedura (vazba v prostředí \mathcal{P}_g)
 $x \longrightarrow 10$ (vazba v prostředí \mathcal{P}_5)
 $y \longrightarrow (x (y))$ (vazba v prostředí \mathcal{P}_5)
 primitivní procedura je aplikována na argumenty 10 a $(x (y))$;
 výsledek: $(10 x (y))$
 výsledek: $(10 x (y))$
 primitivní procedura *list* je aplikována na argumenty $((x (y)) . y)$ a $(10 x (y))$;
 výsledek: $((x (y)) . y) (10 x (y))$
 výsledek: $((x (y)) . y) (10 x (y))$
 výsledek: $((x (y)) . y) (10 x (y))$
 výsledek: $((x (y)) . y) (10 x (y))$
 výsledek: $((x (y)) . y) (10 x (y))$

