

I. Java - Základné elementy



1. Výrazy (Expressions)

- ☐ Premenné
- ☐ Operácie a funkcie
- ☐ Syntaktický strom
- ☐ Vypočítanie hodnoty
- ☐ Pravidlá priority
- ☐ Syntaktické pravidlá

Program (v Jave): Příklad

```
public class Suma
{
    public static int f(int n)
    { return n * ( n + 1) / 2; }

    public static void main(String[] args)
    { int n = Integer.parseInt(args[0]);

        System.out.print(„Výsledok je: „);
        System.out.println( f(n) );
    }
}
}
```



Výraz

Vykonanie programu



Program uložit' v súbore `Suma.java`

```
> javac Suma.java
```

```
> java Suma 25
```

```
Výsledok je: 325
```

```
>
```

Ďalšie príklady (výrazov)

`n * (n + 1) / 2`

`- sum + i`

`a % b` // Zvyšok po delení a / b

`Math.min(a, b)` // Minimum z hodnôt a a b v Jave

`Math.abs(a)` // Absolútna hodnota a v Jave

`Math.abs(Math.min(a % b, c))` // v Jave

Výrazy



Sú vybudované z:

☐ Premenných: `n`, `a`, `sum`

(musia byť deklarované: neskôr)

☐ Operátorov: `+`, `-`, `*`, `/`, `%`

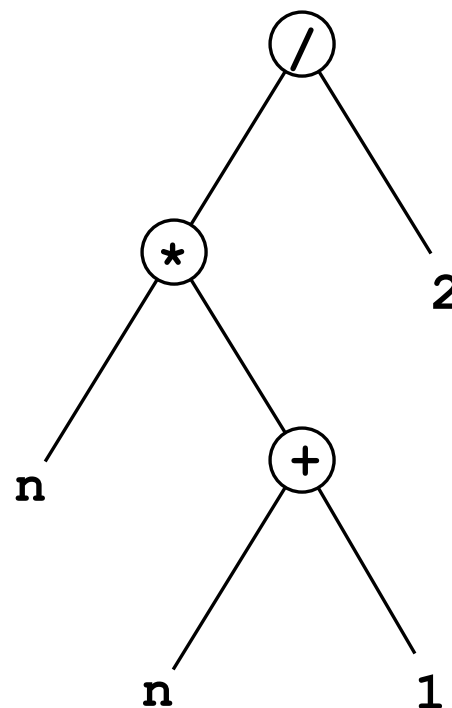
☐ Funkcií: `Math.abs(n)`

☐ Konštánt: `1`, `2`, `0`, `4711`, `...`

☐ Zátvoriek: `()`

Syntaktický strom

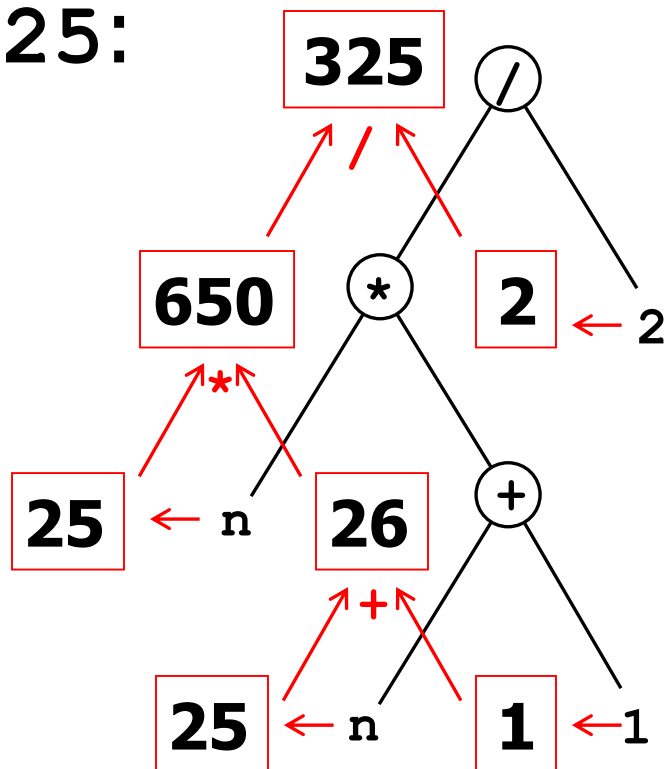
$n * (n + 1) / 2$



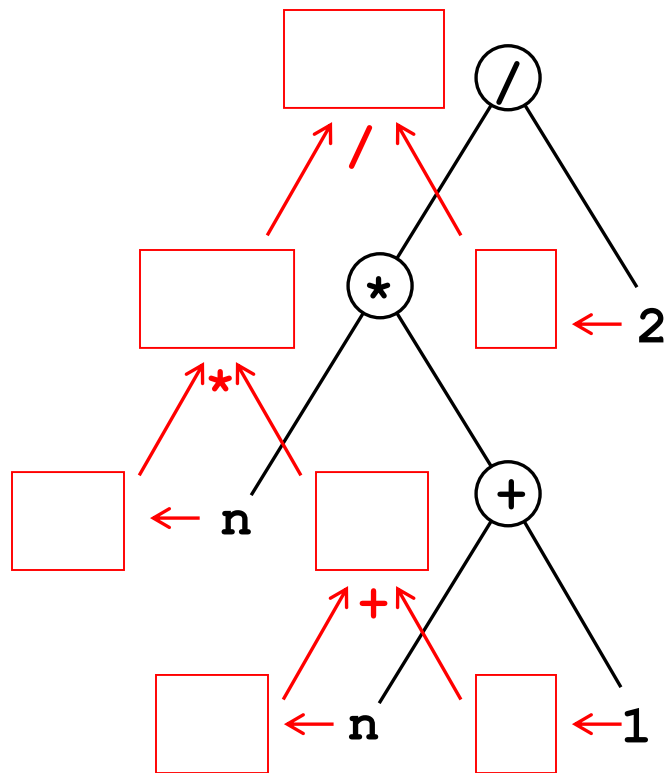
Ku každému výrazu
existuje **jednoznačný**
Syntaktický strom!

Vypočítanie hodnoty

Pre $n = 25$:



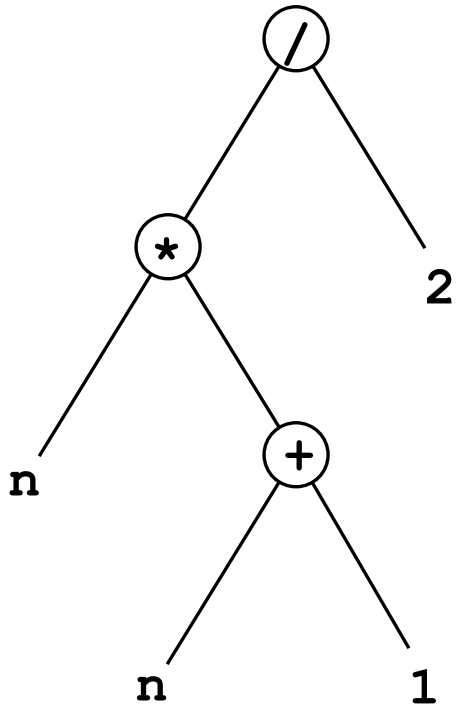
Vyhodnocovací formulár výrazu



Každému výrazu je pomocou syntaktického stromu jednoznačne priradený formulár pre vyhodnotenie výrazu.

Ďalšie príklady

$n * n + 1 / 2$

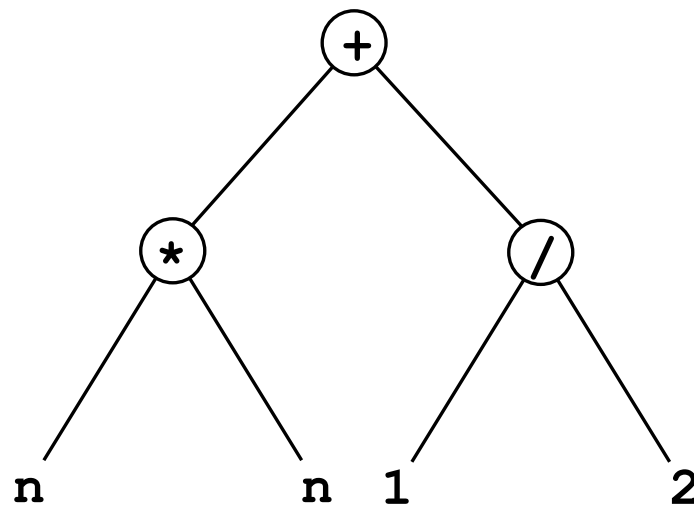
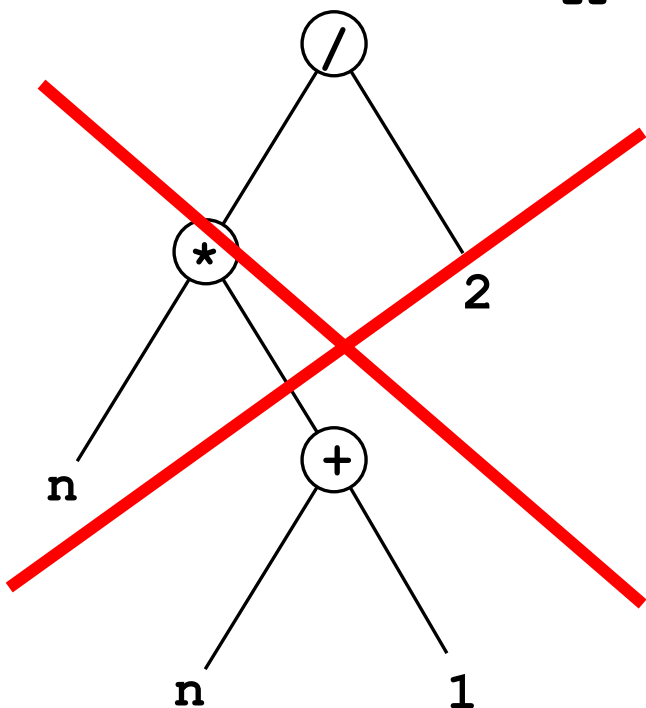


?

Ďalšie príklady

(násobenie a delenie pred sčítaním)

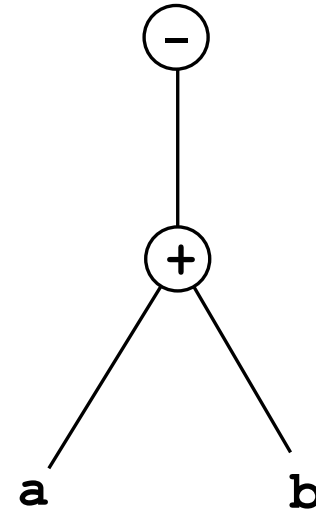
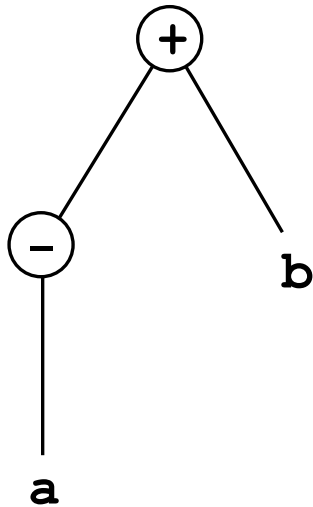
$$n * n + 1 / 2$$



**Násobenie a
delenie pred
sčítaním**

Ďalšie príklady

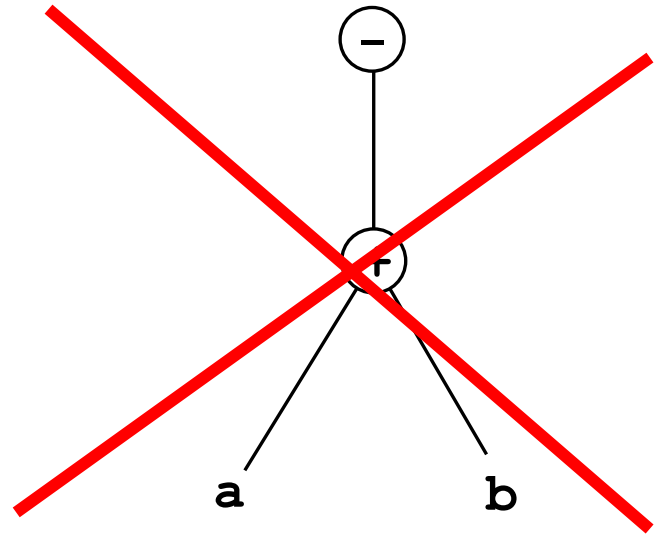
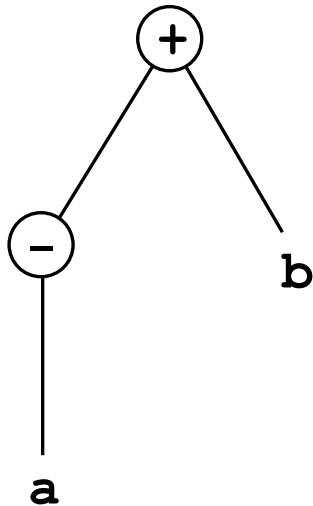
$$- a + b$$



Ďalšie príklady

(unárne operácie pred binárnymi)

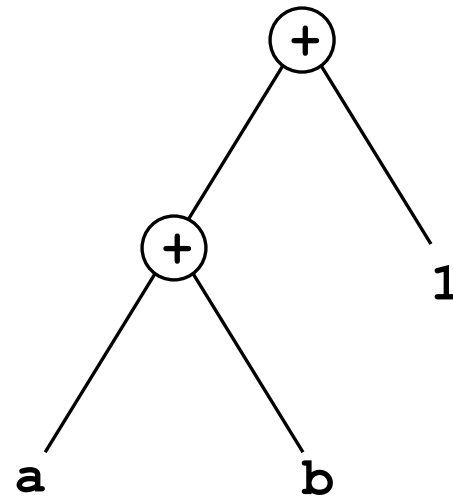
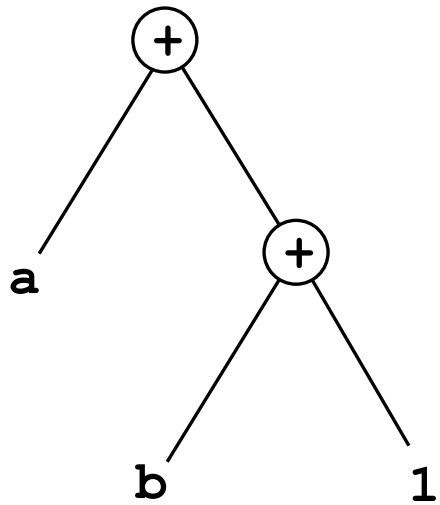
$- a + b$



**unárne pred
binárnymi**

Ďalšie príklady

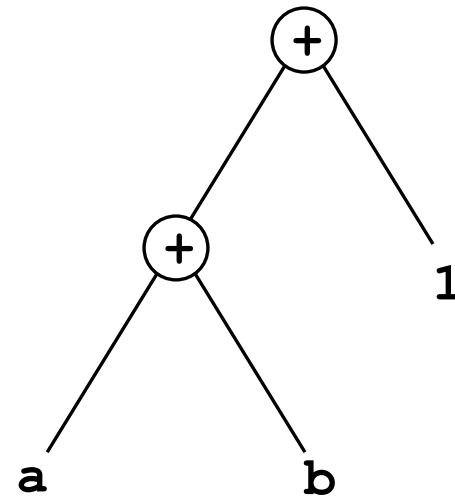
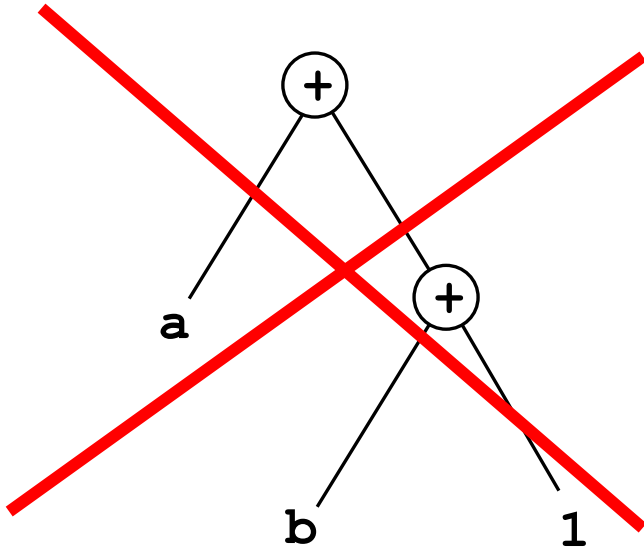
$a + b + 1$



Ďalšie príklady

(zľava doprava)

a + b + 1

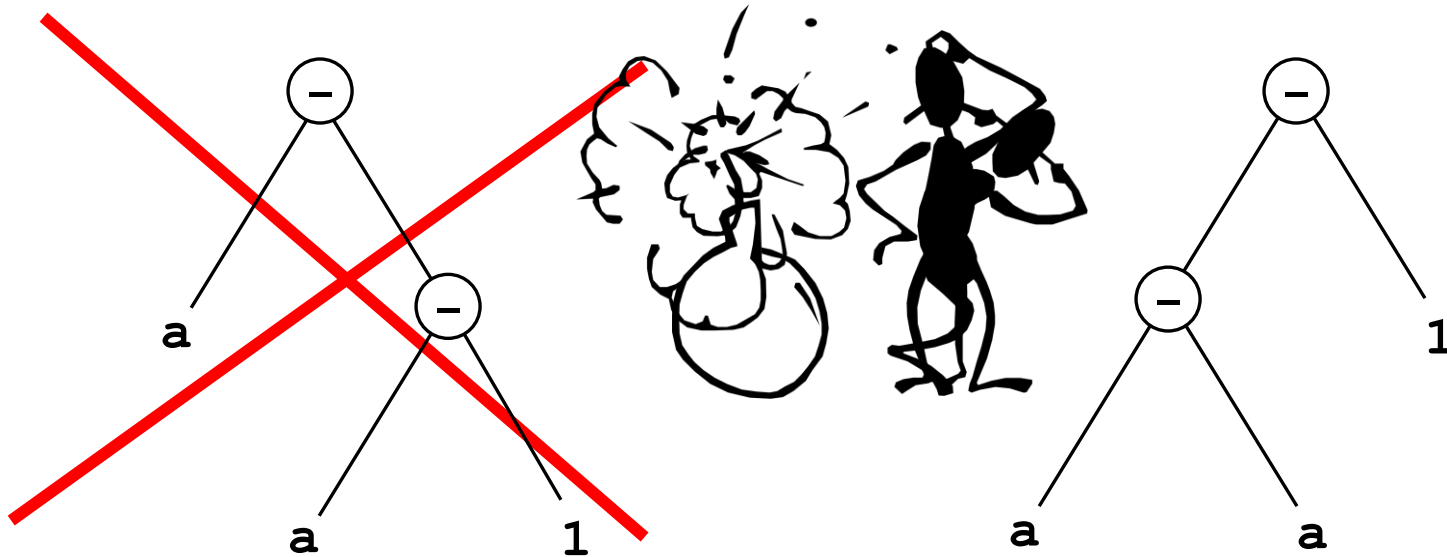


zľava doprava

Ďalšie príklady

(zl'ava doprava)

a - a - 1



zl'ava doprava

Protipříklady (výrazov)

`?n * (n + 1 / 2`

`?- sum § j`

`?) a + b (`

`?Math.min(a;b)`

`? (a + b) * c)`

**Ktoré výrazy sú korektné
(přípustné)?**

Gramatika (Syntaktické pravidlá)

<Výraz> =

<MultVýraz> |

<Výraz> "+" <MultVýraz> |

<Výraz> "-" <MultVýraz>

<MultVýraz> =

<UnárnyVýraz> |

<MultVýraz> "*" <UnárnyVýraz> |

<MultVýraz> "/" <UnárnyVýraz> |

<MultVýraz> "%" <UnárnyVýraz>

Gramatika (pokračovanie)

<UnárnyVýraz> =

<Identifikátor>

<Konštanta>

" (" <Výraz> ") "

<Identifikátor> " (" <ZoznamArgumentov> ") "

"+" <UnárnyVýraz>

"-" <UnárnyVýraz>

opcionálne, t.j.
môže byť
vynechaný

<ZoznamArgumentov> =

<Výraz>

<ZoznamArgumentov> ", " <Výraz>

Gramatika (pokračovanie)

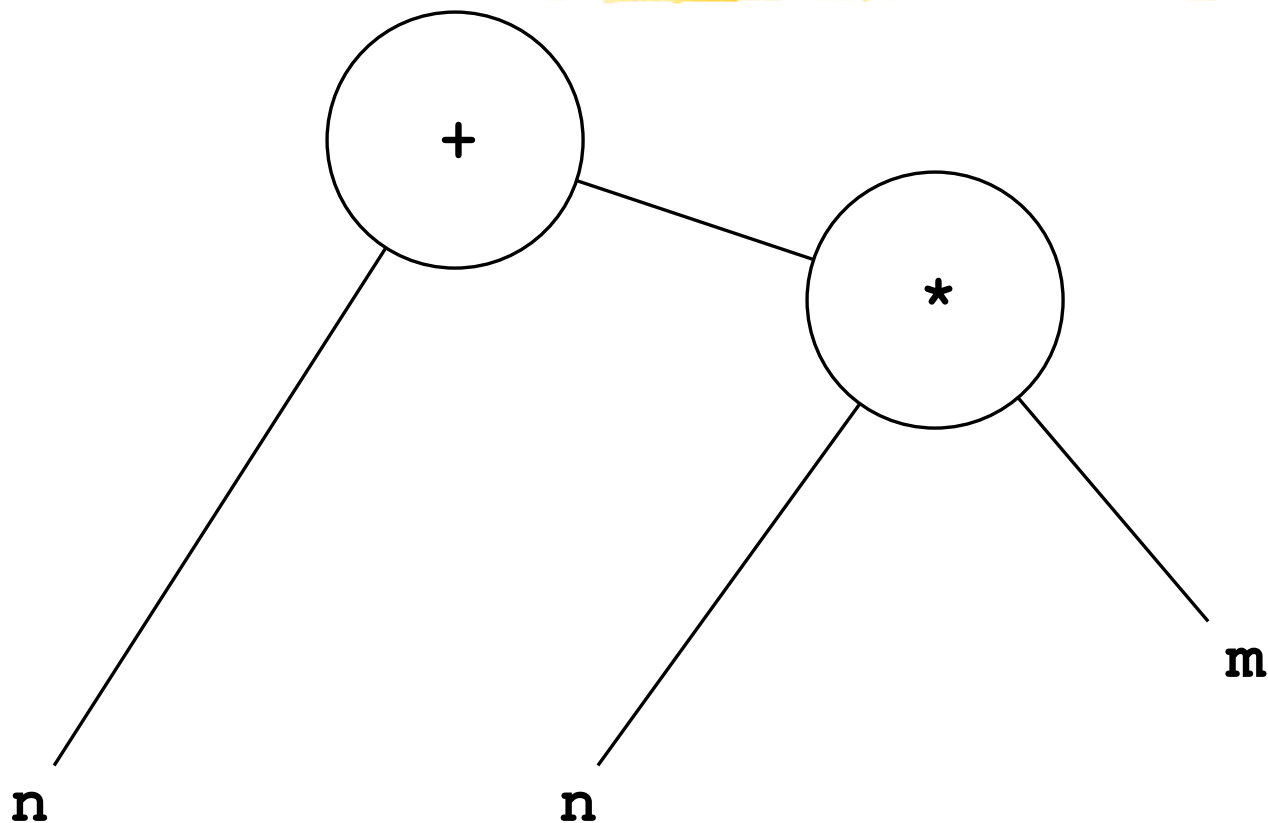
`<Identifikátor> = ...`
`<Konštanta> = ...` } jasné!?

Táto notácia popisujúca syntax sa nazýva **EBNF** (Extended Backus-Naur-Form).

Syntax jazyka Java a C++ je definovaná veľmi podobnou notáciou



Aplikácia pravidiel

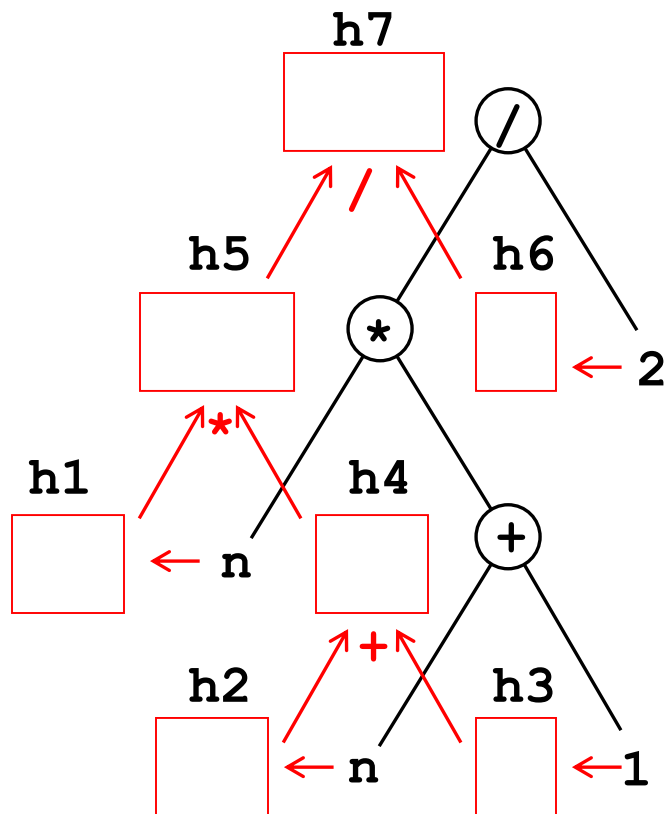


Odvodenie a vyhodnotenie



- ❓ Syntaktické pravidlá (v EBNF) definujú syntakticky prípustné výrazy.
- ❓ Odvodenie každého výrazu je jednoznačné.
- ❓ Výsledkom tohto odvodenia je syntaktický strom výrazu a jeho prostredníctvom vyhodnocovací formulár výrazu.

Vyhodnotenie formuláru



```
int h1 = n;  
int h2 = n;  
int h3 = 1;  
int h4 = h2 + h3;  
int h5 = h1 * h4;  
int h6 = 2;  
int h7 = h5 / h6;  
  
return h7;
```