Základní práce s čísly a výrazy

Posloupnosti

U posloupností se dají snadno počítat jejich nepříliš komplikované limity. Hromadné body limsup
a liminf se u posloupností v tomto programu počítají špatně, je nutné jít přes MaxValue a MinValue a potom použít limity. V mnoha případech program úlohu nezvládne.

Bezproblémové posloupnosti

V první části jsou ukázány příklady, které program spočte dobře. Jsou to všechno situace, kdy limitovaná posloupnost je zúžení vhodné funkce na množinu přirozených čísel a tato funkce má limitu v nekonečnu (např. posloupnost {sin n} je zúžení funkce sin x, $\{2^n\}$ je zúžení funkce 2^x, posloupnost {n!} je zúžení funkce Gamma(x+1) na přirozená čísla). Program počítá limitu posloupnosti jako limitu funkce definované v okolí nekonečna.

```
In[1]:= Limit[a^(1 / n), n → Infinity]
Out[1]= 1

In[2]:= Limit[n^(1 / n), n → Infinity]
Out[2]= 1

In[3]:= Limit[Sin[n] / n, n → Infinity]
Out[3]= 0

In[4]:= Limit[2^n / n!, n → Infinity]
Out[4]= 0

In[5]:= Limit[n / (n!)^(1 / n), n → Infinity]
Out[5]= e

In[6]:= Limit[(1 + a / n)^n, n → Infinity]
Out[6]= e^a
```

Problémové limity

Další část ukazuje příklady, kde sice limitovaná posloupnost je zúžení vhodné funkce na množinu
přirozených čísel ale tato funkce nemá limitu v nekonečnu. Pak se musí přidat podmínka, že se pracuje jen v oboru přirozených čísel. Kromě jednoduchých případù bývá proto vhodnější tuto podmínku přidávat vždy. Jak je vidět z jednoduchého posledního příkladu, ani přidání této podmínky nezajistí výpočet jednoduché limity.

```
In[7]:= Limit[(-1)^(2 * n + 1), n -> Infinity]

        Nonreal::warning : Nonreal number encountered.

Out[7]= Nonreal

In[8]:= Limit[(-1)^(2 * n + 1), n -> Infinity, Assumptions → Element[n, Integers]]
Out[8]= -1
```

In[9]:=
```
Limit[Tan[2 * n * Pi / 2], n -> Infinity, Assumptions → Element[n, Integers]]
```
Out[9]=
```
0
```

In[10]:=
```
Limit[Sin[n ! * Pi], n -> Infinity]
```
Out[10]=
```
Interval[{-1, 1}]
```

In[11]:=
```
Limit[Sin[n ! * Pi], n -> Infinity, Assumptions → Element[n, Integers]]
```
Out[11]=
```
0
```

In[12]:=
```
Limit[Sin[n ! * Pi / 2], n -> Infinity, Assumptions → Element[n, Integers]]
```
Out[12]=
```
Interval[{-1, 1}]
```

Numerický přístup

Může se stát, že program nedovede napsat přesnou hodnotu limity, ale dovede ji odhadnout numericky. K tomu slouží příkaz NLimit (nejdříve je nutné nahrát numerický balíček Numerical Calculus}.

In[13]:=
```
Limit[Sum[1 / (k^3 * Log[k]), {k, 2, n}], n -> Infinity]
```
Out[13]= $\text{Limit}\left[\sum_{k=2}^{n} \frac{1}{k^3\,\text{Log}[k]},\ n \to \infty\right]$

In[14]:=
```
Limit[Sum[1 / (k^3 * Log[k]), {k, 2, n}],
  n -> Infinity, Assumptions → Element[n, Integers]]
```
Out[14]= $\text{Limit}\left[\sum_{k=2}^{n} \frac{1}{k^3\,\text{Log}[k]},\ n \to \infty,\ \text{Assumptions} \to n \in \text{Integers}\right]$

In[15]:=
```
Needs["NumericalCalculus`"]
```
In[16]:=
```
NLimit[Sum[1 / (k^3 * Log[k]), {k, 2, n}], n → Infinity]
```
Out[16]=
```
0.237996
```

Rekurentní posloupnosti

Poslední část se zabývá situací, kdy je posloupnost definována rekurentně. První příklad s konstantní posloupností ukazuje, že si Mathematica s touto situací jednoduše neporadí. Existuje numerický postup (ale nikoli s použitím NLimit), který v některých situacích dá výsledek. Tento výsledek je odhadnut po numerickém výpočtu jistého počtu prvních členů posloupnosti (tento počet se dá ovlivnit). Druhý příklad s výsledkem ($\sqrt{5}$ -1)/2 ukazuje, že výsledek je vypočten velmi přesně. V posledním příkladě je limita rovna -1.

In[17]:=
```
Clear[h]; h[1] = 1; h[n_] := h[n - 1]
```
In[18]:=
```
h[5]
```
Out[18]=
```
1
```
In[19]:=
```
h[100]
```
Out[19]=
```
1
```

```
In[20]:= Limit[h[n], n → Infinity]

$IterationLimit::itlim : Iteration limit of 4096 exceeded. ≫

Out[20]= Limit[Hold[h[(-4095 + n) - 1]], n → ∞]

In[21]:= Limit[h[n], n → Infinity, Assumptions → n ∈ Integers]

$IterationLimit::itlim : Iteration limit of 4096 exceeded. ≫

Out[21]= Limit[Hold[h[(-4095 + n) - 1]], n → ∞, Assumptions → n ∈ Integers]

In[22]:= Limit[h[n], n → Infinity, Assumptions → n ∈ Integers] // N

$IterationLimit::itlim : Iteration limit of 4096 exceeded. ≫

Out[22]= Limit[Hold[h[(-4095. + n) - 1.]], n → ∞, Assumptions → n ∈ Integers]

In[23]:= Needs["NumericalCalculus`"]

In[24]:= NLimit[h[n], n → Infinity, Assumptions → n ∈ Integers]

$IterationLimit::itlim : Iteration limit of 4096 exceeded. ≫

NLimit::notnum : The expression Hold[h[(-4095. + 1.) - 1.]] is not numerical at the point n == 1.`. ≫

Out[24]= NLimit[Hold[h[(-4095 + n) - 1]], n → ∞, Assumptions → n ∈ Integers]
```

```
In[25]:= posl = Table[h[k], {k, 1, 10}]; SequenceLimit[posl]

Out[25]= 1
```

```
In[26]:= Clear[h]; h[1] = 0; h[n_] := h[n] = 1 / (h[n - 1] + 1)

In[27]:= posl = Table[h[k], {k, 1, 10}]; SequenceLimit[posl]

Out[27]= 0.61803398874989484820458683436564

In[28]:= N[(Sqrt[5] - 1) / 2, 32]

Out[28]= 0.61803398874989484820458683436564
```

```
In[29]:= Clear[h]; h[1] = -1 / 2; h[n_] := h[n] = 1 + h[n - 1] + 1 / h[n - 1]

In[30]:= posl = Table[h[k], {k, 1, 10}]; SequenceLimit[posl]

Out[30]= -0.9999999999999999
```