

Třídění, rozdílové seznamy

bubblesort(S, Sorted)

Seznam S setříd'te tak, že

- najděte první dva sousední prvky X a Y v S tak, že $X > Y$, vyměňte pořadí X a Y a získáte S1; a setříd'te S1
- pokud neexistuje žádný takový pár sousedních prvků X a Y, pak je S setříděný seznam

```
swap([X,Y|Rest],[Y,X|Rest1]) :-
    X>Y.                                % nebo obecněji X@>Y pomocí gt(X,Y)
swap([Z|Rest],[Z|Rest1]) :-
    swap(Rest,Rest1).

bubblesort(S,Sorted) :-
    swap (S,S1), !,
    bubblesort(S1, Sorted).
bubblesort(Sorted,Sorted).
```

Hana Rudová, Logické programování I, 27. března 2007

2

Třídění, rozdílové seznamy

quicksort(S, Sorted)

Neprázdný seznam S setříd'te tak, že

- smažte nějaký prvek X z S; rozdělte zbytek S na dva seznamy Small a Big tak, že: v Big jsou větší prvky než X a v Small jsou zbývající prvky
- setříd'te Small do SortedSmall
- setříd'te Big do SortedBig
- setříděný seznam vznikne spojením SortedSmall a [X|SortedBig]

```
quicksort([], []).                    + ošetření případu, kdy S je prázdný seznam
quicksort([X|T], Sorted) :- split(X, Tail, Small, Big),
    quicksort(Small, SortedSmall),
    quicksort(Big, SortedBig),
    append(SortedSmall, [X|SortedBig], Sorted).
```

```
split(X, [], [], []).
split(X, [Y|T], [Y|Small], Big) :- X>Y, !, split(X, T, Small, Big).
split(X, [Y|T], Small, [Y|Big]) :- split(X, T, Small, Big).
```

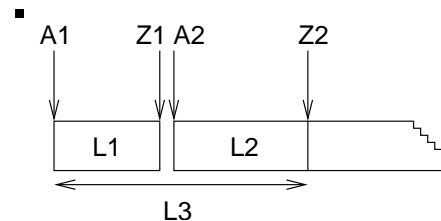
Hana Rudová, Logické programování I, 27. března 2007

3

Třídění, rozdílové seznamy

Rozdílové seznamy

- Zapamatování konce a připojení na konec: rozdílové seznamy
- $[a, b] = L1-L2 = [a, b|T]-T = [a, b, c|S]-[c|S] = [a, b, c]-[c]$
- Reprezentace prázdného seznamu: L-L



- $?- \text{append}([1,2,3|Z1]-Z1, [4,5|Z2]-Z2, S).$
- $\text{append}(A1-Z1, Z1-Z2, A1-Z2).$

L1 L2 L3

Hana Rudová, Logické programování I, 27. března 2007

4

Třídění, rozdílové seznamy

reverse(Seznam, Opacny)

```
reverse( [], [] ).
reverse( [ H | T ], Opacny ) :-
    reverse( T, OpacnyT ),
    append( OpacnyT, [ H ], Opacny ).

reverse( Seznam, Opacny ) :- reverse0( Seznam, Opacny-[] ).
reverse0( [], S-S ).
reverse0( [ H | T ], Opacny-OpacnyKonec ) :-
    reverse0( T, Opacny-[ H | OpacnyKonec ] ).

reverse( Seznam, Opacny ) :- reverse0( Seznam, [], Opacny ).
reverse0( [], S, S ).
reverse0( [ H | T ], A, Opacny ) :-
    reverse0( T, [ H | A ], Opacny ).
```

quicksort pomocí rozdílových seznamů

Neprázdný seznam S setříd'te tak, že

- smažte nějaký prvek X z S;
rozdělte zbytek S na dva seznamy Small a Big tak, že:
v Big jsou větší prvky než X a v Small jsou zbývající prvky
- setříd'te Small do SortedSmall
- setříd'te Big do SortedBig
- setříděný seznam vznikne spojením SortedSmall a [X|SortedBig]

```
quicksort(S, Sorted) :- quicksort1(S,Sorted-[]).
```

```
quicksort([],Z-Z).
```

```
quicksort([X|T], A1-Z2) :-
    split(X, Tail, Small, Big),
    quicksort(Small, A1-[X|A2]),
    quicksort(Big, A2-Z2).
                                append(A1-Z1, Z1-Z2, A1-Z2).
```