Algoritmus navracení (Backtracking Control Strategy)

Recursive procedure **BACKTRACK** (*DATA*)

{*DATA* reprezentuje databázi příslušného stavu řešení}

1. **if TERM** (*DATA*) **then return** *NIL*;

   { **TERM** je logická funkce (predikát) nabývající pravdivostní hodnoty **true** v případě, že obsah databáze *DATA* splňuje cílovou podmínku implementovaného produkčního systému. Jako příznak úspěšného ukončení hledání řešení je procedurou vracen *NIL*, resp. prázdný seznam. }

2. **if DEADEND** (*DATA*) **then return** *FAIL*;

   { **DEADEND** je logická funkce nabývající hodnoty **true** v případě, že mechanismus se dostal do chybového (fail) stavu (podmínky viz výše) a nelze dále úspěšně pokračovat vpřed k hledanému cíli řešení. V takovém případě vrací procedura symbol *FAIL* jako příznak dosažení chybového stavu. }

3. *RULES* ← **APPRULES** (*DATA*);

   { **APPRULES** je funkce určující, která produkční pravidla a v jakém pořadí budou na daný obsah databáze *DATA* aplikována }

4. LOOP: **if NULL** (*RULES*) **then return** *FAIL*;

   { není-li žádné další aplikovatelné pravidlo, končí procedura chybovým (fail) stavem a vrací symbol *FAIL* }

5. *R* ← **FIRST** (*RULES*);

   { aplikováno bude v pořadí první, resp. podle daného kritéria "nejlepší" pravidlo }

6. *RULES* ← **TAIL** (*RULES*);

   { z posloupnosti aplikovatelných produkčních pravidel je vyjmuto zvolené pravidlo }

7. *NEW_DATA* ← **R** (*DATA*);

   { na obsah databáze *DATA* je aplikováno produkční pravidlo **R** z posloupnosti *RULES* a je vygenerován nový obsah databáze *NEW_DATA* }

8. *PATH* ← **BACKTRACK** (*NEW_DATA*);

   { procedura **BACKTRACK** je vyvolána rekursivně pro nový obsah databáze }

9. **if** *PATH* = *FAIL* **then goto** LOOP;

   { vrátilo-li vyvolání procedury **BACKTRACK** chybový příznak *FAIL*, přechází na použití dalšího produkčního pravidla (pokud takové existuje) }

10. **return CONS** (*R, PATH*);

    { v opačném (úspěšném) případě je přidáno nově aplikované produkční pravidlo na čelo seznamu (vrchol zásobníku) reprezentujícího vygenerovanou cestu ve stromu řešení z počátečního uzlu do uzlu aktuálního }