1 Box model

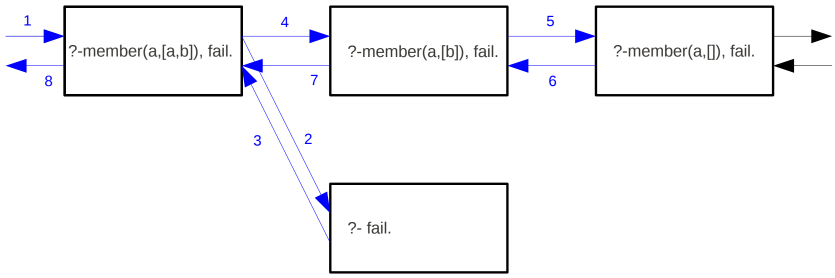
Příklad 1.1: Pro následující program a dotazy předveďte jejich vyhodnocení pomocí box modelu.

```
member(X,[X|_]).
member(X,[_|T]) :- member(X,T).

?- member(a,[b]).
?- member(a,[b,a]).
?- member(a,[a,b]), fail.
```

Řešení 1.1: Zjednodušené (nevnořované) zakreslení boxů a toku výpočtu pro dotaz

```
?- member(a,[a,b]), fail.
```



Kromě nakreslení boxů lze sledovat porty boxů i pomocí trasování přímo v Prologu. Např.:

```
?- trace.
?- member(a,[b,a]).
        1      1 Call: member(a,[b,a]) ?
        2      2 Call: member(a,[a]) ?
        2      2 Exit: member(a,[a]) ?
        1      1 Exit: member(a,[b,a]) ?
yes
```

□

2 Metainterprety, zpětné a dopředné řetězení

Příklad 2.1: Napište metainterpret pro zpětné řetězení

- pro prologovské klauzule s konjunkcí a disjunkcí
- pro pravidla ve tvaru rule(LefthandSide, RighthandSide), kde argument LefthandSide je seznam prologovských predikátů (čárka označuje konjunkci) a RighthandSide je predikát.

Řešení 2.1:

- ```
  prove(true).
  prove((A;B)) :- prove(A) ; prove(B).
  prove((A,B)) :- prove(A), prove(B).
  prove(H) :- clause(H,B), prove(B).
  ```
- ```
  prove([]).
  prove([H|T]) :- prove(H), prove(T).
  prove(RHS) :- rule(LHS,RHS), prove(LHS).
  ```

□

Příklad 2.2: Napište metainterpret pro dopředné řetězení pro pravidla ve tvaru rule(LHS, RHS).

Řešení 2.2:

```
fc(K1,K3) :- step(K1,K2), fc(K2,K3).
fc(K1,K1).

step(K1,K2) :-
 rule(L,R),
 true_in(L,K1),
 not true_in(R,K1),
 append(K1,[R],K2).

true_in([],_).
true_in([H|T],K) :-
 true_in(H,K),
 true_in(T,K).

true_in(X,K) :- member(X,K).
```

□

Příklad 2.3: Přepište níže uvedená pravidla pro metainterpret pro dopředné řetězení a simulujte jeho chování pro následující tři fakta: `c. d. e.`

```
rule( ( a :- b, c )).
rule( ( a :- f, c )).
rule( ( f :- d )).
rule( ( b :- d, e )).
```

Řešení 2.3:

```
rule([b,c],a).
rule([f,c],a).
rule([d],f).
rule([d,e],b).

?- fc([c,d,e],X).
```

□

3 SAT: Davis Putnam (DP), DPLL

Příklad 3.1: Pomocí algoritmu DP ověřte splnitelnost množiny klauzulí

$$S = \{\{P,Q,R\},\{P,\neg Q,\neg R\},\{P,\neg W\},\{\neg Q,\neg R,\neg W\},\{\neg P,\neg Q,R\},\\ \{U,X\},\{U,\neg X\},\{Q,\neg U\},\{\neg R,\neg U\}\}$$

Řešení 3.1: Algoritmus DP (jedna z variant):

- vstup: formule v CNF bez tautologií (reprezentovaná množinově)
- opakuj, dokud jsou proměnné a v množině není $\square$:
  - vyber proměnnou
  - vytvoř všechny rezolventy klauzulí obsahujících tuto proměnnou, přidej do zpracovávané množiny ty z nich, které nejsou tautologie
  - vyřaď ze zpracovávané množiny všechny klauzule obsahující tuto proměnnou
- výstup:
  - SAT v případě, že zpracovávaná množina zůstane prázdná (NO CLAUSES),
  - UNSAT v případě, že některá rezolventa je $\square$ (EMPTY CLAUSE)

a) výběr proměnné: W
klauzule obsahující: $\{P,\neg W\},\{\neg Q,\neg R,\neg W\}$
rezolventy: žádné
nová množina: $\{\{P,Q,R\},\{P,\neg Q,\neg R\},\{\neg P,\neg Q,R\},\{U,X\},\{U,\neg X\},\{Q,\neg U\},\{\neg R,\neg U\}\}$

b) výběr proměnné: X
klauzule obsahující: $\{U,X\},\{U,\neg X\}$
rezolventy: $\{U\}$
nová množina: $\{\{P,Q,R\},\{P,\neg Q,\neg R\},\{\neg P,\neg Q,R\},\{U\},\{Q,\neg U\},\{\neg R,\neg U\}\}$

c) výběr proměnné: U
klauzule obsahující: $\{U\},\{Q,\neg U\},\{\neg R,\neg U\}$
rezolventy: $\{Q\},\{\neg R\}$
nová množina: $\{\{P,Q,R\},\{P,\neg Q,\neg R\},\{\neg P,\neg Q,R\},\{Q\},\{\neg R\}\}$

d) výběr proměnné: Q
klauzule obsahující: $\{P,Q,R\},\{P,\neg Q,\neg R\},\{\neg P,\neg Q,R\},\{Q\}$
rezolventy: $\{\neg P,R\},\{P,\neg R\},\{P,R,\neg R\},\{P,\neg P,R\}$
z toho tautologie: $\{P,R,\neg R\},\{P,\neg P,R\}$
nová množina: $\{\{\neg R\},\{\neg P,R\},\{P,\neg R\}\}$

e) výběr proměnné: P
klauzule obsahující: $\{\neg P,R\},\{P,\neg R\}$
rezolventy: $\{R,\neg R\}$
z toho tautologie: $\{R,\neg R\}$
nová množina: $\{\{\neg R\}\}$

f) výběr proměnné: R
klauzule obsahující: $\{\neg R\}$
rezolventy: žádné
nová množina: prázdná (NO CLAUSES)

Závěr: původní množina je splnitelná. $\square$

Příklad 3.2: Pomocí algoritmu DPLL ověřte splnitelnost množiny klauzulí

$$S = \{\{P, \neg Q, \neg R\}, \{R, \neg Q\}, \{\neg P, \neg Q\}, \{P, \neg R\}, \{P, R\}, \{R\}, \{Q, \neg P, Q\}\}$$

Řešení 3.2: Algoritmus DPLL: dokud nelze uplatnit pravidlo SAT nebo UNSAT, aplikuj na množinu klauzulí S následující pravidla:

- UNSAT: S obsahuje $\square$
- SAT: S je prázdná
- MULT: eliminace duplicitních literálů v klauzuli
- SUBS: vynechání nadmnožiny jiné klauzule
- UNIT: vynechání $\neg L$ ve všech klauzulích, pokud S obsahuje $\{L\}$
- TAUT: vynechání tautologické klauzule (obsahující L i $\neg L$)
- PURE: vynechání všech klauzulí obsahujících L, pokud se $\neg L$ v S nevyskytuje
- SPLIT: vyřazení všech klauzulí obsahujících L nebo $\neg L$ a přidání všech jejich rezolvent

a) $\{\{P, \neg Q, \neg R\}, \{R, \neg Q\}, \{\neg P, \neg Q\}, \{P, \neg R\}, \{P, R\}, \{R\}, \{Q, \neg P, Q\}\}$ MULT

b) $\{\{P, \neg Q, \neg R\}, \{R, \neg Q\}, \{\neg P, \neg Q\}, \{P, \neg R\}, \{P, R\}, \{R\}, \{\neg P, Q\}\}$ SUBS

c) $\{\{P, \neg Q, \neg R\}, \{R, \neg Q\}, \{\neg P, \neg Q\}, \{P, \neg R\}, \{R\}, \{\neg P, Q\}\}$ UNIT R

d) $\{\{P, \neg Q\}, \{R, \neg Q\}, \{\neg P, \neg Q\}, \{P\}, \{R\}, \{\neg P, Q\}\}$ PURE R

e) $\{\{P, \neg Q\}, \{\neg P, \neg Q\}, \{P\}, \{\neg P, Q\}\}$ UNIT P

f) $\{\{P, \neg Q\}, \{\neg Q\}, \{P\}, \{Q\}\}$ PURE P

g) $\{\{\neg Q\}, \{Q\}\}$ SPLIT Q

h) $\{\square\}$ UNSAT

Závěr: původní množina je nesplnitelná.

$\square$