

## Knut Bendix procedura

Pre definisanja same procedure uvodimo pojam prezapisivanja. Prezapisivanje predstavlja korišćenje transformacija termova na osnovu usmerenih jednakosti (npr.  $A + A \rightarrow 2A$ ).

Ponekad sistemi za prezapisivanje imaju svojstvo da se svi ekvivalentni termovi svode na istu normalnu formu, tj. na term na koji nije dalje moguće primeniti ni jedno pravilo. Dok postoje sistemi koji nemaju svojstvo zaustavljanja ( $1+2 \rightarrow 2+1 \rightarrow 1+2 \dots$ ) i za takve sisteme kažemo da su nekonfluentni.

Knut Bendix procedura nam omogućava da u nekim slučajevima nekonfluentan sistem dopunimo određenim termovima i da nakon toga on postane konfluentan.

Novi pojam koji se nameće jesu kritični parovi. Neka su  $l_1 \rightarrow r_1$  i  $l_2 \rightarrow r_2$  dva pravila koja nemaju zajedničkih promenljivih. Neka je  $l_1'$  podterm terma  $l_1$  koji nije varijabla i neka je  $O$  najopstiji unifikator termova  $l_1'$  i  $l_2$ . Term  $l_1[l_1' \rightarrow O(l_2)]$  određuje kritični par  $\langle O(r_1), O(l_1)[O(l_1') \rightarrow O(r_2)] \rangle$ .

Sama procedura se zasniva na teoremi koja kaže da je sistem za prezapisivanje lokalno konfluentan ako i samo ako su mu svi kritični parovi povezivi. Tj, ako važi da za kritični par  $\langle u_1, u_2 \rangle$  nije poveziv, tj. Postoje normalne forme ovih termova  $u_1'$  i  $u_2'$  koje nisu jednake. Dodavanjem pravila  $u_1' \rightarrow u_2'$  ili  $u_2' \rightarrow u_1'$  sistem se može učiniti konfluentnim.

Indeks klasa:

**class** Signature;

Signatura se sastoji iz funkcijskih i predikatskih simbola kojima su pridružene arnosti (nenegativni celi brojevi).

**class** Structure;

L-struktura.

**class** Valuation;

Valuacija.

**class** BaseTerm;

Apstraktna klasa BaseTerm koja predstavlja termine.

**class** VariableTerm;

Term koji predstavlja jednu varijablu.

**class** FunctionTerm;

Term koji predstavlja funkcijski simbol primenjen na odgovarajući broj podtermova.

**class** BaseFormula;

Apstraktna klasa kojom se predstavljaju formule.

```
class AtomicFormula;
```

Klasa predstavlja sve atomicke formule (True, False i Atom).

```
class LogicConstant;
```

Klasa predstavlja logicke konstante (True i False).

```
class False;
```

Klasa predstavlja logicku konstantu False.

```
class True;
```

Klasa predstavlja True logicku konstantu.

```
class Atom;
```

Klasa predstavlja atom.

```
class UnaryConnective;
```

Klasa unarni veznik (obuhvata negaciju).

```
class Not;
```

Klasa koja predstavlja negaciju.

```
class BinaryConnective;
```

Klasa predstavlja sve binarne veznike.

```
class And;
```

Klasa predstavlja konjunkciju.

```
class Or;
```

Klasa predstavlja disjunkciju.

```
class Imp;
```

Klasa predstavlja implikaciju.

```
class Iff;
```

Klasa predstavlja ekvivalenciju.

```
class Quantifier;
```

Klasa predstavlja kvantifikovane formule.

```
class Forall;
```

Klasa predstavlja univerzalno kvantifikovanu formulu.

```
class Exists;
```

Klasa predstavlja egzistencijalnog kvantifikatora.

**class** Function;

Apstraktni tip podatka kojim se predstavlja funkcija  $D^n \rightarrow D$  kojom se mogu interpretirati funkcijski simboli arnosti  $n$ .

**class** Relation;

Apstraktni tip podataka kojim se predstavlja relacija  $D^n \rightarrow \{0,1\}$  kojom se mogu interpretirati predikatski simboli arnosti  $n$ .

Funkcije:

Term **getSubterm** (**const** Term &t);

Funkcija koja odredjuje prvi podterm koji nije varijabla.

**void** **getAllCriticalPairs** (CriticalPair &criticals, Formula f1, Formula f2, Signature &s);

Funkcija za odredjivanje kritičnih parova.

**void** **knut\_bendix** (RewriteSystem &system, Signature &s);

Funkcija koja primenjuje Knut Bendix proceduru upotpunjavanja.