DENOTACIJSKA SEMANTIKA - ZGLEDI

Zgled 1:
Definiraj sintakso in semantiko preprostih boolovih izrazov.V izrazih lahko nastopajo sprem. x, y in z(ki imajo nedefinirano vrednost true ali false), konstante true in false ter operatorji and, or in not.
Nekaj primerov sintakticno pravilnih izrazov:

z
x or (y and not z)
not x and (true and (false or y))

1. Sintakticna domena(v BNF notaciji):

<exp> ::=
<fact> and <exp> | <fact> or <exp> |
<fact>
<fact> ::=
true | false | <var>
not <fact> | (<exp>)
<var> ::= x | y | z

2. Opis stanja in semanticnih funkcij:

State ≡ {true, false} *x* {true, false} *x* {true, false}
// stanje opisujejo vrednosti sprem. x, y in z
Mexp: exp -> (State -> { true, false })
// izraz preslika trenutno stanje sprem. x, y, in z v vrednost izraza true ali false
Mfact: fact -> (State -> { true, false })
MVar: Var -> (State -> { true, false })

3.Definicija semanticnih funkcij:

Mexp [e:exp] (s:State) ≡
<fact> and <exp> => Mfact[e.fact](s) AND Mexp[e.exp](s) |
<fact> or <exp> => Mfact[e.fact](s) OR Mexp[e.exp](s) |
<fact> => Mfact[e.fact](s)
Mfact [f:fact] (s:State) ≡
true => true |
false => false |
<var> => Mvar[f.var](s) |
not <fact> => NOT Mfact[f.fact](s) |
(<exp>) => Mexp[f.exp](s)
Mvar [v:var] (<xVal, yVal, zVal>:State) ≡
x => xVal |
y => yVal |
z => zVal

Zgled 2:

Denimo da je vrednost stanja State = <true, true, true>, in je pomen boolovega izraza definiran kot v resitvi zgleda 1.Kaksen je potem pomen izraza: x or (y and not z) ?
Torej:

Mexp[x or (y and not z)] (<true, true, true>) = ?

Resitev:

Mexp[x or (y and not z)] (<true, true, true>) =
Mfact[x] (<true, true, true>) OR Mexp[(y and not z)] (<true, true, true>) =
Mvar[x] (<true, true, true>) OR Mexp[(y and not z)] (<true, true, true>) =
true OR Mexp[(y and not z)] (<true, true, true>) =
true OR Mfact[(y and not z)] (<true, true, true>) =
true OR Mexp[y and not z] (<true, true, true>) =
true OR (Mfact[y](<true, true, true>) AND Mexp[not z](<true, true, true>)) =
true OR (Mvar[y](<true, true, true>) AND Mexp[not z](<true, true, true>)) =
true OR (true AND Mfact[not z](<true, true, true>)) =
true OR (true AND (NOT Mfact[z](<true, true, true>)))=
true OR (true AND (NOT Mvar[z](<true, true, true>)))=
true OR (true AND (NOT true)) =
true OR (true AND false) =
true OR false =
true

Zgled 3:
Gornjo gramatiko(zgled 1) za boolove izraze razsirimo s prireditvenim in sestavljenim stavkom, tako da je sintakticno pravilen program:

```
program
  x := true ;
  y := false ;
  z := false ;
  x := x and y ;
  y := x or ( true and (not z) )
end
```

Ustrezno definiraj sintakso(v BNF notaciji) ter denotacijsko semantiko programa !

1. Sintakticna domena(v BNF notaciji):

```
<prog> ::=
        program <statements> end
<statements> ::=
        <lvar> := <exp> ; <statements> |
        <lvar> := <exp>
// <lvar> ima sint. enak pomen kot <var>, vendar drugacen semanticni pomen !
<lvar> ::=
        x | y | z

// <exp> isto kot v Zgledu 1
<exp> ::=
        <fact> and <exp> | <fact> or <exp> |
        <fact>
<fact> ::=
        true | false | <var>
        not <fact> | ( <exp> )
<var> ::=
        x | y | z
```

2. Opis stanja in semanticnih funkcij:

State ≡ {true, false} *x* {true, false} *x* {true, false}
// stanje opisujejo vrednosti sprem. x, y in z
Mprog: prog -> (State -> State)
Mstatements: statements -> (State -> State)
Mlvar: lvar -> {1, 2, 3}
// <lvar> preslika ime spremenljivke v njen index v stanju
Mexp: exp -> (State -> { true, false })
// izraz preslika trenutno stanje sprem. x, y, in z v vrednost izraza true ali false
Mfact: fact -> (State -> { true, false })
Mvar: var -> (State -> { true, false })

3.Definicija semanticnih funkcij:

```
Mprog [p:prog] (s:State) ≡
        program <statements> end => Mstatements[p.statements](s)

Mstatements [sts:statements] (<s1, s2, s3>:State) ≡
        <lvar> := <exp> ; <statements> =>
                CASE Mlvar[sts.lvar] OF
                        1 => Mstatements[sts.statements](<Mexp[sts.exp](s), s2, s3>)
                        2 => Mstatements[sts.statements](<s1, Mexp[sts.exp](s), s3>)
                        3 => Mstatements[sts.statements](<s1, s2, Mexp[sts.exp](s)>)
                END
        <lvar> := <exp> =>
                CASE Mlvar[sts.lvar] OF
                        1 => <Mexp[sts.exp](s), s2, s3>
                        2 => <s1, Mexp[sts.exp](s), s3>
                        3 => <s1, s2, Mexp[sts.exp](s)>
                END

Mlvar [lv:lvar] ≡
        x => 1 |
        y => 2 |
        z => 3

Mexp [e:exp] (s:State) ≡
        <fact> and <exp> => Mfact[e.fact](s) AND Mexp[e.exp](s) |
        <fact> or <exp> => Mfact[e.fact](s) OR Mexp[e.exp](s) |
        <fact> => Mfact[e.fact](s)

Mfact [f:fact] (s:State) ≡
        true => true |
        false => false |
        <var> => Mvar[f.var](s) |
        not <fact> => NOT Mfact[f.fact](s) |
        ( <exp> ) => Mexp[f.exp](s)

Mvar [v:var] (<xVal, yVal, zVal>:State) ≡
        x => xVal |
        y => yVal |
        z => zVal
```