

## Prevođenje programskih jezika - Oktobar 2011.

1. Napisati program koji omogućava korisniku rad sa iskaznim formulama.

- (a) Iskazne formule se dobijaju rekurzivno od logičkih promenljivih (čija imena se sastoje od proizvoljnog broja malih slova engleske abecede i cifara), logičkih konstanti **True** i **False**, konačnom primenom konjunkcije, disjunkcije, implikacije, ekvivalencije i negacije nad već definisanim formulama. Formule mogu da sadrže i zagrade. Napraviti interpreter koji proverava sintaksnu ispravnost formula koje se unose sa ulaza. [8]

```
p /\ q <=> r
!p => q \/ r <=> s
p1 /\ (p2 <=> !p3)
p /\ /\ r                                     syntax error
```

- (b) Interpreteru napravljenom u delu pod a, dodati naredbu dodele: [8]

```
p := True
q := False
r := p <=> q
```

- (c) Unaprediti interpreter tako da računa vrednost zadate formule: [8]

```
p := True
q := False
r := p <=> q
p /\ q <=> r <=> True           True
!p => q \/ r <=> s             Variable s is not defined.
```

- (d) Omogućiti da u slučaju korišćenja promenljive koja dotada nije definisana, program ipak sračuna vrednost formule, na taj način što će dati konkretnu vrednost (**True** ili **False**) ako vrednost formule ne zavisi od neinicijalizovane promenljive, a inače je vrednost formule nova logička konstanta **Undef**. [6]

```
p := True
q := False
p /\ q \/ s           Undef
p \/ s \/ q           True
!(q => s)              False
```

## Prevođenje programskih jezika - Oktobar 2011.

1. Napisati program koji omogućava korisniku rad sa iskaznim formulama.

- (a) Iskazne formule se dobijaju rekurzivno od logičkih promenljivih (čija imena se sastoje od proizvoljnog broja malih slova engleske abecede i cifara), logičkih konstanti **True** i **False**, konačnom primenom konjunkcije, disjunkcije, implikacije, ekvivalencije i negacije nad već definisanim formulama. Formule mogu da sadrže i zagrade. Napraviti interpreter koji proverava sintaksnu ispravnost formula koje se unose sa ulaza. [8]

```
p /\ q <=> r
!p => q \/ r <=> s
p1 /\ (p2 <=> !p3)
p /\ /\ r                                     syntax error
```

- (b) Interpreteru napravljenom u delu pod a, dodati naredbu dodele: [8]

```
p := True
q := False
r := p <=> q
```

- (c) Unaprediti interpreter tako da računa vrednost zadate formule: [8]

```
p := True
q := False
r := p <=> q
p /\ q <=> r <=> True           True
!p => q \/ r <=> s             Variable s is not defined.
```

- (d) Omogućiti da u slučaju korišćenja promenljive koja dotada nije definisana, program ipak sračuna vrednost formule, na taj način što će dati konkretnu vrednost (**True** ili **False**) ako vrednost formule ne zavisi od neinicijalizovane promenljive, a inače je vrednost formule nova logička konstanta **Undef**. [6]

```
p := True
q := False
p /\ q \/ s           Undef
p \/ s \/ q           True
!(q => s)              False
```