Computational Logic, Toets 1

(23 maart 2009, 10.00-12.00)

Zet op ieder vel tentamenpapier Uw naam en Collegekaartnummer. Geef bij berekeningen ook tussenresultaten. De opgaven tellen allemaal voor 10 punten, behalve 2 en 1 die voor 15 tellen.

Opgave 1. Laat met resolutie zien dat de laatste zin een conclusie is uit de voorafgaande zinnen. Vermeldt alle substituties.

(1) Paarden en ezels zijn rijdieren.

(2) Iedereen die een rijdier heeft rijdt erop.

(3) Sjonny heeft een ezel.

(4) Sjonny rijdt op een ezel.

Opgave 2. Schrijf in clausule vorm:

1. $(p \wedge q) \vee (\neg q \wedge r)$
2. $p \leftrightarrow (r \wedge s)$
3. $\forall x \exists z[\exists y(A(x,y) \wedge B(y,z)) \rightarrow \exists x C(x,z)]$

Opgave 3. Er zijn verschillende strategiën om resolutie te bedrijven. Geef van elk van de volgende strategiën in een paar zinnen een karakterisatie. Noem bij elk ook een voordeel en nadeel, en geef daarbij een korte toelichting.

1. Backward chaining
2. Unit preference
3. Subsumption

Opgave 4. Beschouw de verzameling formules $S = \{(A \wedge C) \rightarrow B, C \rightarrow \neg D, (A \wedge D) \rightarrow \neg B, B \rightarrow D\}$.

1. Bepaal RC(S).
2. Bepaal aan de hand van RC(S) of S vervulbaar is.
3. Als S vervulbaar is, contrueer dan een model volgens het algoritme van de completeness stelling.

Opgave 5. Hieronder staat in pseudocode een algoritme voor resolutie.

function PL-RESOLUTION(KB, α) **returns** *true* or *false*
inputs: KB: the knowledge base, α: the query,
    both a sentence in propositional logic
clauses ← the set of clauses
        in the CNF representation of KB $\wedge \neg\alpha$
new ← {}
loop do
  for each C_i, C_j **in** *clauses* **do**
    resolvents ← PL-RESOLVE(C_i, C_j)
    if *resolvents* contains the empty clause
      then return *true*
    new ← *new* ∪ *resolvents*
  if *new* ⊆ *clauses* **then return** *false*
  clauses ← *clauses* ∪ *new*

De output van PL-RESOLVE is de verzameling van alle clauses verkregen door resolutie op zijn inputs.

1. Geef een voorbeeld waarbij de output *true* is. Geef precies aan wat dan (dus aan het eind) de waarde is van *resolvents* en van *clauses* .
2. Geef een voorbeeld waarbij de output *false* is. Geef precies aan wat dan (dus aan het eind) de waarde is van *resolvents* en van *clauses* .

Opgave 6. Beschouw de volgende verzameling clauses:
$\{\{A(a,x), \neg B(y, f(x))\}, \{\neg B(g(c,x), z)\}\}$.

1. Geef het Herbrand universum. Als het groot is, hoeft U niet meer dan 12 elementen te geven, maar dan wel zo dat U de verschillende typen laat zien.
2. Geef een Herbrand interpretatie waarin deze verzameling niet vervulbaar is.
3. Zeg in een paar zinnen wat het belang is van Herbrand interpretaties voor de theorie van resolutie.

Opgave 7. Beschouw de volgende twee uitspraken:

a) Als $\models \alpha \wedge \beta$ dan $\models \alpha$ en $\models \beta$

b) Als $\models \alpha \vee \beta$ dan $\models \alpha$ of $\models \beta$

1. Toon aan dat uitspraak a) juist is.

2. Toon aan dat uitspraak b) niet juist is.

Opgave 8. We onderzoeken een afleidingsregel die we de naam *Modus Snenop* geven. Een toepassing van de regel levert twee formules. Als α en β formules uit de propositielogica zijn, dan levert toepassing van de regel op een verzameling S met $\alpha \in S$, de formules $\beta \rightarrow \alpha$ en β op. Deze twee formules worden vervolgens aan S toegevoegd.

We definieëren dat $S \vdash_{MS} \gamma$ geldt als door een eindig aantal Modus-Snenop stappen γ afgeleid kan worden uit S (waarbij dus ook de afgeleide formules mogen worden gebruikt).

1. Wat betekent het als we zeggen dat Modus Snenop *sound* is?

2. Wat betekent het als we zeggen dat Modus Snenop *complete* is?

3. Toon aan dat Modus Snenop *niet* sound is.

4. Toon aan dat Modus Snenop complete is.