Artificiële Intelligentie
Project 1

Philippe Dellaert
3e bachelor Computerwetenschappen-Elektrotechniek

18 april 2007

Hoofdstuk 1

Alergie voorbeeld

1.1 Inleiding

We beginnen met het allergie voorbeeld vanuit de cursus te herbekijken en een vergelijking te maken tussen twee volgordes van oplossen. Bij de eerste volgorde zullen eerst de positieve voorbeelden verwerkt worden en daarna pas de negatieve voorbeelden. We verwachten dat er een snellere convergentie is. De tweede volgorde is die waar de negatieve voorbeelden eerst verwerkt worden. We verwachten dat dit zorgt voor een grotere complexiteit, wat er dan weer voor zorgt dat alles trager is. We verwachten uiteraard ook dat beide volgordes dezelfde oplossing geven.

1.2 Positieve voorbeelden eerst

- **Initiële situatie**
  $G = \{(?,?,?,?)\}$
  $S = \{\bot\}$
- **+ (Alma 3, breakfast, friday, cheap)**
  $G = \{(?,?,?,?)\}$
  $S = \{(\text{Alma 3,breakfast,friday,cheap})\}$
- **+ (Alma 3, lunch, saterday, cheap)**
  $G = \{(?,?,?,?)\}$
  $S = \{(\text{Alma 3,?,?,cheap})\}$
- **- (De Moete, lunch, friday, expensive)**
  $G = \{(\text{Alma 3,?,?,?}),(?,?,?,\text{cheap})\}$
  $S = \{(\text{Alma 3,?,?,cheap})\}$
- **- (Sedes, breakfast, sunday, cheap)**
  $G = \{(\text{Alma 3,?,?,?})\}$
  $S = \{(\text{Alma 3,?,?,cheap})\}$
- **- (Alma 3, breakfast, sunday, expensive)**
  $G = \{(\text{Alma 3,?,?,cheap})\}$
  $S = \{(\text{Alma 3,?,?,cheap})\}$

1.3 Negatieve voorbeelden eerst

- **Initiële situatie**
  $G = \{(?,?,?,?)\}$
  $S = \{\perp\}$

- **- (De Moete, lunch, friday, expensive)**
  $G = \{$(?,breakfast,?,?),(?,?,?,cheap),(?,?,saturday,?),
  (?,?,sunday,?),(almadrie,?,?,?),(sedes,?,?,?)$\}$
  $S = \{\perp\}$

- **- (Sedes, breakfast, sunday, cheap)**
  $G = \{$(?,breakfast,?,expensive),(?,breakfast,friday,?),(?,lunch,?,cheap),(?,lunch,sunday,?),
  (?,?,friday,cheap),(?,?,saturday,?),(?,?,sunday,expensive),(Alma 3,?,?,?),
  (De Moete,breakfast,?,?),(De Moete,?,?,cheap),(De Moete,?,sunday,?),(Sedes,lunch,?,?),
  (Sedes,?,?,expensive),(Sedes,?,friday,?)$\}$
  $S = \{\perp\}$

- **- (Alma 3, breakfast, sunday, expensive)**
  $G = \{$(?,breakfast,friday,?),(?,lunch,?,cheap),(?,lunch,sunday,?),(?,?,friday,cheap),
  (?,?,saturday,?),(Alma 3,lunch,?,?),(Alma 3,?,?,cheap),(Alma 3,?,friday,?),
  (De Moete,breakfast,?,?),(De Moete,?,?,cheap),(De Moete,?,sunday,?),(Sedes,lunch,?,?),
  (Sedes,?,?,expensive),(Sedes,?,friday,?)$\}$
  $S = \{\perp\}$

- **+ (Alma 3, breakfast, friday, cheap)**
  $G = \{$(?,breakfast,friday,?),(?,?,friday,cheap),(Alma 3,?,?,cheap),(Alma 3,?,friday,?)$\}$
  $S = \{$(Alma 3,breakfast,friday,cheap)$\}$

- **+ (Alma 3, lunch, saterday, cheap)**
  $G = \{$(Alma 3,?,?,cheap)$\}$
  $S = \{$(Alma 3,?,?,cheap)$\}$

1.4 Vergelijking

Zoals verwacht geven beide methodes hetzelfde resultaat als de voorbeelden in de cursus. Het grote verschil echter, ligt in de complexiteit van de oplossing. Indien men eerst de positieve voorbeelden verwerkt gaat dit veel sneller dan wanneer men de negatieve voorbeelden eerst verwerkt. Dit blijkt ook duidelijk uit de tijdsmetingen van beide methodes als ze 40 maal worden uitgevoerd gebruik makend van het Version spaces programma.

Tijdsmeting voor positieve voorbeelden eerst:
Used total time : 881 milliseconds.
Used total inferences : 284160

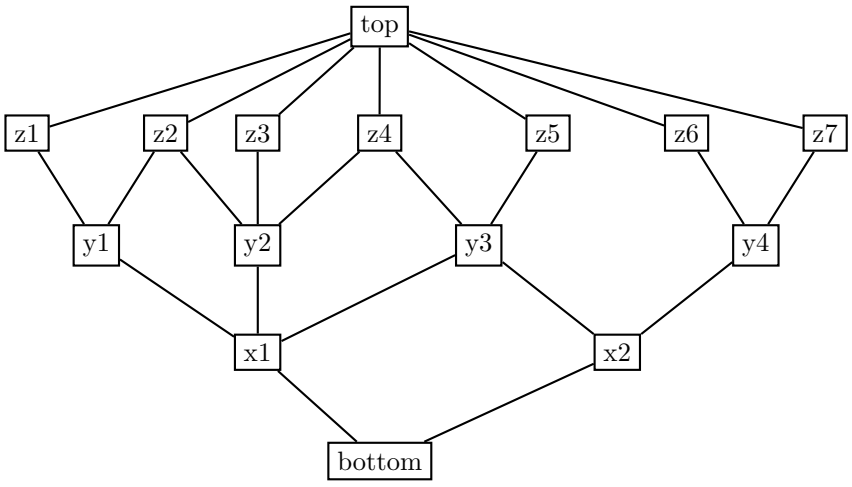
Tijdsmeting voor negatieve voorbeelden eerst:
Used total time : 4586 milliseconds.
Used total inferences : 4737240

Hoofdstuk 2

Dualiteit

2.1 Inleiding

In de hiërarchie die hier gebruikt wordt, zijn er drie niveau's tussen het top en het bottom element. Op het eerste niveau staan zeven elementen z1 tot z7. Deze elementen hebben top als parent. Op het tweede niveau staan vier elementen, y1 tot y4. y1 heeft de elementen z1 en z2 als parent. y2 heeft de elementen z2, z3 en z4 als parent. y3 heeft de elementen z4 en z5 als parent. y4 heeft z6 en z7 als parent. Het derde niveau bestaat uit twee elementen, x1 en x2. x1 heeft y1, y2 en y3 als parent. x2 heeft y3 en y4 als parent. Daaronder volgt nog bottom, die x1 en x2 als parent heeft. Deze structuur staat ook uitgebeeld in het diagramma.



De concepten zelf staan op het eerste niveau en elk concept komt overeen met de ontkenning van een event. Dit geeft dus een verzameling van de volgende events: { z1_event, z2_event, z3_event, z4_event, z5_event, z6_event, z7_event }. Er zal gewerkt worden met de volgende voorbeelden:

- + z1_event
- - z2_event
- - z4_event
- + z5_event
- + z7_event

Indien dit wordt opgelost, zou dit ons moeten leiden naar y2.

De code die is toegevoegd aan de verschillende bestanden van de Version Spaces Tool, is bijgevoegd in een appendix.

2.2 Positieve voorbeelden eerst

- **Initiële situatie**
  $G = \{(?)\}$
  $S = \{\bot\}$
- **+ z1_event**
  $G = \{(?)\}$
  $S = \{(x2),(y2)\}$
- **+ z5_event**
  $G = \{(?)\}$
  $S = \{(y2),(y4)\}$
- **+ z7_event**
  $G = \{(?)\}$
  $S = \{(y2),(z6)\}$
- **- z2_event**
  $G = \{(z2)\}$
  $S = \{(y2)\}$
- **- z4_event**
  $G = \{(y2)\}$
  $S = \{(y2)\}$

2.3 Negatieve voorbeelden eerst

- **Initiële situatie**
  $G = \{(?)\}$
  $S = \{\bot\}$
- **- z2_event**
  $G = \{(z2)\}$
  $S = \{\bot\}$
- **- z4_event**
  $G = \{(y2)\}$
  $S = \{\bot\}$

- **+ z1_event**
  $G = \{(y2)\}$
  $S = \{(y2)\}$
- **+ z5_event**
  $G = \{(y2)\}$
  $S = \{(y2)\}$
- **+ z7_event**
  $G = \{(y2)\}$
  $S = \{(y2)\}$

2.4 Vergelijking

Ook hier geven beide methodes hetzelfde resultaat, wat vanzelfsprekend is. Echter valt wel direct op dat de tweede methode veel sneller convergeert naar een oplossing dan de tweede. Dit blijkt ook uit de tijdsmetingen wanneer we elke methode 40 maal na elkaar uitvoeren in de Version Spaces Tool.

Tijdsmeting voor positieve voorbeelden eerst:
Used total time : 803 milliseconds.
Used total inferences : 113640

Tijdsmeting voor negatieve voorbeelden eerst:
Used total time : 361 milliseconds.
Used total inferences : 99120

We kunnen er algemeen van uitgaan dat het het interessantst is om eerst de voorbeelden te verwerken die de meeste beperkingen opleggen. Concreet betekent dit dat als we een hiërarchie hebben waarbij we een branching van top naar bottom hebben, we best de positieve voorbeelden eerst verwerken, zoals in hoofdstuk 1 van dit project. Hebben we echter een hiërarchie waarbij er branching is van bottom naar top, dan kunnen we beter eerst de negatieve voorbeelden verwerken.

Als we zouden werken met meerdere attributen, dan zouden we normaal dezelfde veronderstelling kunnen maken. Zeker als alle attributen dezelfde branching richting hebben. De complexiteit in zijn geheel zal wel toenemen aangezien er meer mogelijkheden zijn die moeten onderzocht worden.

Bijlage A

Code

A.1 vs_1.pl

```
son(z1,abstract_top).
son(z2,abstract_top).
son(z3,abstract_top).
son(z4,abstract_top).
son(z5,abstract_top).
son(z6,abstract_top).
son(z7,abstract_top).
son(y1,z1).
son(y1,z2).
son(y2,z2).
son(y2,z3).
son(y2,z4).
son(y3,z4).
son(y3,z5).
son(y4,z6).
son(y4,z7).
son(x1,y1).
son(x1,y2).
son(x1,y3).
son(x2,y3).
son(x2,y4).
son(abstract_bottom,x1).
son(abstract_bottom,x2).

nmap(z1, z1_event).
nmap(z2, z2_event).
nmap(z3, z3_event).
nmap(z4, z4_event).
nmap(z5, z5_event).
nmap(z6, z6_event).
nmap(z7, z7_event).
```

A.2 exampledata.pl

```
exampledata(abstract_concept,
    ['[z1_event]:p','[z2_event]:n','[z4_event]:n','[z5_event]:p','[z7_event]:p']).
```