Administrivia

• Homework 4
  – graded
  – you should get it back today
Administrivia

• this Thursday
  – computer lab class
  – help with homework 5
  – meet in SS 224
Today’s Topics

• Homework 4 Review

• Finish with
  – Chapter 6: Quantifiers
  – Quiz 5 (end of class)
Homework 4 Review

• Questions 1 and 2
• Worlds

– w1 → \{A,B\}  
  horse(a). horse(b).
– w2 → \{B,C\}  
  horse(b). horse(c).
– w3 → \{A,B,C\}  
  horse(a). horse(b). horse(c).
– w4 → \emptyset  
  :- dynamic horse/1.

  or

– w5 → \{A,B,C,D,E\} 
  horse(a). horse(b). horse(c). horse(d).
  or
  :- assert(horse(a)), retract(horse(a)).
  or
  :- set_prolog_flag(unknown,fail).
– w6 → \{A,B,C,D,E,F\} 
  horse(a). horse(b). horse(c). horse(d).
  or
  horse(e).
  or
  horse(e). horse(f).
Homework 4 Review

• Prolog definitions common to worlds $W_1,..,W_6$:

  - `horses(Sum) :-`
  - `findall(X,horse(X),L),`
  - `sum(L,Sum).`

  - `sum(L,X+Y) :- pick(X,L,Lp), pick(Y,Lp,_)`. 
  - `sum(L,X+Sum) :- pick(X,L,Lp), sum(Lp,Sum).`

  - `pick(X,[X|L],L).`
  - `pick(X,[_|L],Lp) :- pick(X,L,Lp).`
Homework 4 Review

- **Questions 1 and 2**
- **Answers to the query**
  - `findall(PL,horses(PL),L), length(L,N).`
  - `w1` → `{A,B}`
    - `L = [a+b]`
    - `N=1`
  - `w2` → `{B,C}`
    - `L = [b+c]`
    - `N=1`
  - `w3` → `{A,B,C}`
    - `L = [a+b,a+c,b+c,a+(b+c) N=4`
  - `w4` → `∅`
    - `L = []`
    - `N=0`
  - `w5` → `{A,B,C,D,E}`
    - `L = [a+b,a+c,a+d,a+e,b+c,b+d,b+e,c+d,c+e,d+e, a+(b+c),a+(b+d),a+(b+e),a+(c+d),a+(c+e),a+(d+e), a+(b+(c+d)),a+(b+(c+e)),a+(b+(d+e)), a+(b+(c+(d+e))),a+(c+(d+e)),b+(c+d), b+(c+e),b+(d+e),b+(c+(d+e)),c+(d+e)]`
    - `total: 26`
  - **2 horses** = \( \binom{5}{2} = \frac{5!}{2!(5-2)!} = 10 \)
  - **3 horses** = \( \binom{5}{3} = \frac{5!}{3!(5-3)!} = 10 \)
  - **4 horses** = \( \binom{5}{4} = \frac{5!}{4!(5-4)!} = 5 \)
  - **5 horses** = 1
Homework 4 Review

- Questions 1 and 2
- Answers to the query
  - `?- findall(PL,horses(PL),L), length(L,N).`
  - `w6 -> {A,B,C,D,E,F}`
  - `L=[a+b,a+c,a+d,a+e,a+f,b+c,b+d,b+e,b+f,c+d,c+e,c+f,d+e,d+f,e+f,`
  - `a+(b+c),a+(b+d),a+(b+e),a+(b+f),a+(c+d),a+(c+e),a+(c+f),a+(d+e),a+(d+f),a+(e+f),`
  - `a+(b+c+d),a+(b+(c+e)),a+(b+(c+f)),a+(b+(d+e)),a+(b+(d+f)),a+(b+(e+f)),`
  - `a+(b+(c+(d+e))),a+(b+(c+(d+f)),a+(b+(c+(e+f))],`
  - `a+(b+(c+(d+(e+f)))), a+(b+(d+(e+f))),`
  - `a+(c+(d+e)),a+(c+(d+f)),a+(c+(e+f)),`
  - `a+(c+(d+(e+f))], a+(d+(e+f)),`
  - `b+(c+d),b+(c+e),b+(c+f),b+(d+e),b+(d+f),b+(e+f),`
  - `b+(c+(d+e)),b+(c+(d+f)),b+(c+(e+f)),`
  - `b+(c+(d+(e+f))], b+(d+(e+f)),`
  - `c+(d+e),c+(d+f),c+(e+f),c+(d+(e+f)),d+(e+f)]`

2 horses = \( \binom{6}{2} = \frac{6!}{2!4!} = 15 \)
3 horses = \( \binom{6}{3} = \frac{6!}{3!3!} = 20 \)
4 horses = \( \binom{6}{4} = \frac{6!}{4!2!} = 15 \)
5 horses = \( \binom{6}{5} = \frac{6!}{5!1!} = 6 \)
6 horses = \( \binom{6}{1} = 1 \)
Total: \( 57 \)
Homework 4 Review
Question 3:
- What is the Prolog query for “three horses”?

Answer
- Notice all cases of threes are of pattern/form _+( _+_ ) where each _ represents an individual horse
  - e.g. a+(b+c), a+(b+d), a+(b+e), a+(c+d), a+(c+e), a+(d+e)

Query (1st attempt)
- ?- findall(PL,(horses(PL),PL=_+( _+_),L).

Example (W_5)
- L=[a+(b+c), a+(b+d), a+(b+e), a+(c+d), a+(c+e), a+(d+e), a+(b+(c+d)), a+(b+(c+e)), a+(b+(d+e)), a+(b+(c+(d+e))), a+(c+(d+e)), b+(c+d), b+(c+e), b+(d+e), b+(c+(d+e)), c+(d+e)]
- Total: 16 (but answer should be 10!)
Homework 4 Review

• **Question 3:**
  - What is the Prolog query for “three horses”?

• **Answer**
  - **Query (2nd attempt)**
    - `?- findall(PL,(horses(PL),PL=+(PL,H),\+H=+(L),L),length(L,N)).`
  - **Example (W5)**
    - `L=[a+(b+c),a+(b+d),a+(b+e),a+(c+d),a+(c+e),a+(d+e),b+(c+d),b+(c+e),b+(d+e),c+(d+e)]`
    - `N=10 (correct)`
Homework 4 Review

• **Question 3:**
  – What is the Prolog query for “three horses”?

• **Another way to deal with the question:**
  – recognize that notation `Horse+Sum` from the given definition:
    – `sum(L, X+Y) :- pick(X, L, Lp), pick(Y, Lp, _).`
    – `sum(L, X+Sum) :- pick(X, L, Lp), sum(Lp, Sum).`
  – is isomorphic to [Head|Tail] list notation (here: + is equivalent to |)

• Write a recursive length predicate, call it `len/2`, for `Horse+Sum`
  – `len(_,Sum,N) :- !, len(Sum,M), N is M+1.`
  – `len(_,1).`

• **Query becomes:**
  – `?- findall(PL, (horses(PL), len(PL,3)), L).`
Question 4:

- How would you write the query for “the three horses”?

Clue (given in lecture slides)

- `?- findall(X,dog(X),List), length(List,1).`
- encodes the definite description “the dog”
  - i.e. query holds (i.e. is true) when dog(X) is true and there is a unique X in a given world

Combine this clue with the answer to Question 3

Resulting Query

- `?- findall(PL,(horses(PL),PL=_+( _+H),\+H=_+_),L), length(L,1).`
- Under the assumption that everything is equally salient, query is true for world W₃ only!
- L = [a+(b+c)]
- Worlds W₁, W₂ and W₄ have too few horses, and worlds W₅ and W₆ have too many.
Back to Chapter 6
Negative Polarity Items

- **Negative Polarity Items (NPIs)**
- **Examples:**
  - ever, any
- **Constrained distribution:**
  - have to be *licensed* in some way
  - grammatical in a "negated environment" or "question"

- **Examples:**
  - (13a) Shelby won’t ever bite you
  - (13b) Nobody has any money
  - (14a) *Shelby will ever bite you
  - (14b) *Noah has any money
  - * = ungrammatical
  - (15a) Does Shelby ever bite?
  - (15b) Does Noah have any money?
Negative Polarity Items

• Inside an *if-clause*:
  – (16a) *If* Shelby *ever* bites you, I’ll put him up for adoption
  – (16b) *If* Noah has *any* money, he can buy some candy

• Inside an *every-NP*:
  – (17a) *Every* dog which has *ever* bitten a cat feels the admiration of other dogs
  – (17b) *Every* child who has *any* money is likely to waste it on candy

• Not inside a *some-NP*:
  – (17a) *Some* dog which has *ever* bitten a cat feels the admiration of other dogs
  – (17b) *Some* child who has *any* money is likely to waste it on candy

Not to be confused with free choice (FC) *any* (meaning: ∀): *any man can do that*
Downwards and Upwards Entailment (DE & UE)

- class $\not\subseteq$ super-class

**Example:**
- hyponym $\not\subseteq$ hypernym
- dog $\not\subseteq$ animal
- Keeshond $\not\subseteq$ dog

**Inferencing:**
- non-negative sentence: **upwards**
  - (23) I have a dog \hspace{1cm} \textbf{(entails)}
  - (23b) I have an animal
  - I have a Keeshond \hspace{1cm} \textbf{(invalid inference)}
- negative sentence: **downwards**
  - (24a) I don’t have a dog \hspace{1cm} \textbf{(entails)}
  - (24b) I don’t have a Keeshond
  - I don’t have an animal \hspace{1cm} \textbf{(invalid inference)}
Downwards and Upwards Entailment (DE & UE)

• **Quantifier every** has semantics
  - \( \{X: P_1(X)\} \subseteq \{Y: P_2(Y)\} \)
  - e.g. every woman likes ice cream
  - \( \{X: \text{woman}(X)\} \subseteq \{Y: \text{likes}(Y, \text{ice\_cream})\} \)

• **Every** is DE for \( P_1 \) and UE for \( P_2 \)
• Examples:
  - (25) a. Every dog barks
  - b. Every Keeshond barks (valid)
  - c. Every animal barks (invalid)
    - semantically, “Keeshond” is a sub-property or subset with respect to the set “dog”
Downwards and Upwards Entailment (DE & UE)

- Quantifier every has semantics
  - \{X: P_1(X)\} \subseteq \{Y: P_2(Y)\}
  - e.g. every woman likes ice cream
  - \{X: \text{woman}(X)\} \subseteq \{Y: \text{likes}(Y, \text{ice_cream})\}
- Every is DE for P_1 and UE for P_2

Examples:
- (25) a. Every dog barks
  - d. Every dog barks loudly (invalid)
  - c. Every dog makes noise (valid)
    - semantically, “barks loudly” is a subset with respect to the set “barks”, which (in turn) is a subset of the set “makes noise”
Downwards and Upwards Entailment (DE & UE)

- **Inferencing:**
  - non-negative sentence: UE
  - (23) I have a dog  \textit{entails} (23b) I have an animal
  - (23b) I have a Keeshond \textit{invalid}
  - negative sentence: DE
  - (24a) I don’t have a dog \textit{entails}
  - (24b) I don’t have a Keeshond
  - I don’t have an animal \textit{invalid}

- **NPI-Licensing:**
  - non-negative sentence: UE
  - (14a) *Shelby will ever bite you
  - (14b) *Noah has any money
  - negative sentence: DE
  - (13a) Shelby won’t ever bite you
  - (13b) Nobody has any money

**Generalization:**
NPIs like \textit{ever} and \textit{any} are licensed by DE
Downwards and Upwards Entailment (DE & UE)

- Inside an every-NP:
  - (17a) [Every [dog][which has ever bitten a cat]] feels the admiration of other dogs
  - (17b) [Every [child][who has any money]] is likely to waste it on candy
- Explanation:
  - every is DE for $P_1$ and UE for $P_2$
  - $\{X: P_1(X)\} \subseteq \{Y: P_2(Y)\}$
- Inside an every-NP:
  - (17a) $P_1 = [dog][which has ever bitten a cat]$
  - (17b) $P_1 = [child][who has any money]$

Generalization:
NPIs like ever and any are licensed by DE
Quiz 5

• Question 1: Is Some UE or DE for $P_1$ and $P_2$?
  – Lecture 22 (Homework 5 Question 3)
    • some: $\{X: P_1(X)\} \cap \{Y: P_2(Y)\} \neq \emptyset$
  – Justify your answer using examples of valid/invalid inferences starting from
    • Some dog barks

• Question 2: Is No UE or DE for $P_1$ and $P_2$?
  – Lecture 22 (Homework 5 Question 3)
    • no: $\{X: P_1(X)\} \cap \{Y: P_2(Y)\} = \emptyset$
  – Use
    • No dog barks