**William Daniels**

**Linguistic Geometry, HW1**

Problem 1: In this problem, the trick is to simply change the false condition (4 instead of 3), and then the rest simply falls out. There are a few special cases to worry about, but they’re few and far between.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| L | ***Q*** | Kernel | ***FT*** | ***FF*** |
|  |  |  |  |  |
| 1 | ***Q*1** | ***S***(c, m) —> ***A***(c + 1, m + 1) | 2 | ∅ |
|  |  |  |  |  |
| 2 | ***Q*2** | ***A***(c, m) —> ***p***(c, m)***A***(f1(c, m), f2(m)) | 2 | 3 |
| 3 | ***Q*3** | ***A***(c, m) —> ***p***(c, m) | ∅ | ∅ |

**All steps of this grammar take place when the boat is on side B of the river.**

**Pred** = { ***Q*1, *Q*2, *Q*3**}

***Q*1**=**T**

***Q*2**(c, m) =**T**if c > 0 or m > 0; ***Q*2**(c, m) =**F**if ((c=4)∧(m = 4)) or

(c > m) ∨ ((4 – c) > (4 – m))

***Q*3**(c, m) =**T**if ((c = 3)∧(m = 3))

**Var** = {c, m}

Fcon = {f1, f2}

f1(c, m) is defined as follows:

If (c = 1) ∧ (m = 1) f1(c, m) = c+ 2

else

if (c ≠ 1) ∧ ( m ≠ 3)

f1(c, m) = c – 1 else

f1(c, m) = c + 1. f2(m) is defined as follows:

If m = 1

f2(m) = m – 1 else

if m = 0

f2(m) = m + 2 else

if m = 4

f2(m) = m else

f2(m) = m + 1

**At the beginning of derivation: c = 0, m = 0**

10

***S***(0, 0)1=> ***A***(1, 1)

2=> ***p***(1, 1)***A***(3, 0)

2=> ***p***(1, 1)***p***(3, 0)***A***(2, 2)

2=> ***p***(1, 1)***p***(3, 0)***p***(2, 2)***A***(1, 3)

2=> ***p***(1, 1)***p***(3, 0)***p***(2, 2)***p***(1, 3)***A***(2, 3)

2=> ***p***(1, 1)***p***(3, 0)***p***(2, 2)***p***(1, 3)***p***(2, 3)***A***(3, 3) 3=> ***p***(1, 1)***p***(3, 0)***p***(2, 2)***p***(1, 3)***p***(2, 3)***p***(3, 3)

Problem 1, part b: The same logic can be applied to the general case, if n is odd, then you can simply use the original controlled grammar, just with all the 3’s replaced by ‘n’s. In the even case, you simply use the aforementioned grammar, and then replace all the4’s with ‘n’s. Very Straightforward. Assume the boat can hold n-1 people.

Problem 2: The towers of Hanoi problem :

**Controlled grammar generating solutions to the Tower of Hanoi Problem**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***L*** | ***Q*** | **Kernel, π*k*** | | **π*n*** | ***FT*** | ***FF*** |
| 1 | ***Q*1** | ***S***(n, x, y) | –> ***A***(n, x, y) |  | 2 | ø |
|  |  |  |  |  |  |  |
| 2 | ***Q*2** | ***A***(n, x, y) | –> ***A***(***f*1**(n), x, ***f*2**(x, y)) |  | 2 | 3 |
|  |  |  | ***p***(n, x, y) |  |  |  |
|  |  |  | ***A***(***f* 1**(n), ***f*2**(x, y), y) |  |  |  |
|  |  |  |  |  |  |  |
| 3 | ***Q*3** | ***A***(n, x, y) | –> ***p***(n, x, y) |  | 2 | ø |

Here ***VT*** ={***p***}

***VN*** ={***S***, ***A***}

***VPR***

*Pred* ={***Q*1***,****Q*2***,****Q*3**},

***Q*1**= ***T***

***Q*2**(n) = ***T***, if n > 1; ***Q*2**(n) = ***F***, if n = 1. ***Q*3**(n) = ***T*,**if n = 1; ***Q*3**(n) = ***F***, if n > 1.

*Var* = {n, x, y}

*F* =*Fcon* ∪ *Fvar*,

*Fcon* = {***f*1**,***f*2**}

***f*1**(n) = n-1, n = 2, 3,...

***f*2**(x, y) yields the value from {***a***, ***b***, ***c***} – {x, y}, where values of

x, y are from {***a***, ***b***, ***c***} *Fvar* = {3,***a***,***c***}

***E*** =**Z+**∪{***a***, ***b***, ***c***}

***Parm***: ***S*** –>*Var*, ***A*** –>*Var*, ***p*** –>*Var* ***L***={1, 2, 3}

**At the beginning of derivation**: x = ***a***, y = ***c***, n = 3.

13

**Derivation of a solution in case of n = 3:**

***S***(3, ***a***, ***c***)1=> ***A***(3, ***a***, ***c***)2=> ***A***(2, ***a***, ***b***)***p***(3, ***a***, ***c***)***A***(2, ***b***, ***c***)

2=> ***A***(1, ***a***, ***c***)***p*** (2, ***a***, ***b*** )***A***(1, ***c***, ***b*** )***p*** (3, ***a***, ***c***)***A***(2, ***b***, ***c***) 3=> ***p***(1, ***a***, ***c***)***p***(2. ***a***, ***b***)***A***(1, ***c***, ***b***)***p***(3, ***a***, ***c***)***A***(2, ***b***, ***c***)

3=> ***p***(1, ***a***, ***c***)***p***(2, ***a***, ***b***)***p***(1, ***c***, ***b***)***p***(3, ***a***, ***c***)***A***(2, ***b***, ***c***)

2=>***p***(1, ***a***, ***c***)***p***(2, ***a***, ***b***)***p***(1, ***c***, ***b***)***p***(3, ***a***, ***c***)***A***(1, ***b***, ***a***)***p***(2, ***b***, ***c***)***A***(1, ***a***, ***c***) 3=> ***p***(1, ***a***, ***c***)***p***(2, ***a*** , ***b***)***p***(1, ***c***, ***b***)***p***(3, ***a*** , ***c***)***p***(1, ***b*** , ***a***)***p***(2, ***b*** , ***c***)***A***(1, ***a***, ***c***) 3=> ***p***(1,***a***, ***c***)***p***(2,***a***, ***b***)***p***(1, ***c***, ***b***)***p***(3, ***a***, ***c***)***p***(1, ***b***, ***a***)***p***(2, ***b***, ***c***)***p***(1, ***a***, ***c***).