

Code.txt

Code.pl

x

```

1  /*Jacob Alspaw*/
2  /*jaa134*/
3  /*EECS 302 Discrete Mathematics*/
4  /*Carl Entemann*/
5
6  /*=====*/
7  /*=====*/
8  /*||      Objects      ||*/
9  /*||  Structs dictating related objects.  ||*/
10 /*=====*/
11 /*=====*/
12
13 /*declaring soups*/
14 soup(
15     black_bean,
16     clam_chowder,
17     mushroom,
18     split_pea,
19     veggie_orzo
20 ).
21
22 /*declaring entres*/
23 entre(
24     chicken_tandoori,
25     fried_steak,
26     italian_sausage,
27     roasted_pork,
28     turkey_burger
29 ).
30
31 /*declaring desserts*/
32 dessert(
33     apple_pie,
34     bundt_cake,
35     cookies,
36     cupcakes,
37     pudding_parfait
38 ).
39
40
41
42 /*=====*/
43 /*=====*/
44 /*||      Rules      ||*/
45 /*||  A variety of given project rules.  ||*/
46 /*=====*/
47 /*=====*/
48
49 /*A test for soups*/
50 isSoup(S) :-
51     S = black_bean;
52     S = clam_chowder;
53     S = mushroom;
54     S = split_pea;
55     S = veggie_orzo.
56
57 /*A test for entres*/
58 isEntre(E) :-
59     E = chicken_tandoori;
60     E = fried_steak;
61     E = italian_sausage;
62     E = roasted_pork;
63     E = turkey_burger.
64
65 /*A test for desserts*/
66 isDessert(D) :-
67     D = apple_pie;
68     D = bundt_cake;
69     D = cookies;
70     D = cupcakes;
71     D = pudding_parfait.

```

```

73  /*A general definition of rule 1.*/
74  /*This rule is too general to apply in individual cases.*/
75  rule1(S, E, D) :-
76      (
77          S \== clam_chowder,
78          E \== roasted_pork,
79          D \== bundt_cake
80      );
81      (
82          (
83              (S \== clam_chowder, E \== roasted_pork, D = bundt_cake);
84              (S \== clam_chowder, E = roasted_pork, D \== bundt_cake);
85              (S = clam_chowder, E \== roasted_pork, D \== bundt_cake)
86          ),
87          servedTuesday(clam_chowder, _, _),
88          servedThursday(_, _, bundt_cake),
89          servedMonday(_, roasted_pork, _)
90      );
91      (
92          (
93              (S \== clam_chowder, E \== roasted_pork, D = bundt_cake);
94              (S \== clam_chowder, E = roasted_pork, D \== bundt_cake);
95              (S = clam_chowder, E \== roasted_pork, D \== bundt_cake)
96          ),
97          servedWednesday(clam_chowder, _, _),
98          servedFriday(_, _, bundt_cake),
99          servedTuesday(_, roasted_pork, _)
100     ).
101
102  /*A special test to denote rule 2.*/
103  rule2(S, E) :-
104      (
105          S = veggie_orzo,
106          E = turkey_burger
107      );
108      (
109          S \== veggie_orzo,
110          E \== turkey_burger
111      ).
112
113  /*A special test to denote rule 3.*/
114  rule3(S, E1, E2, D) :-
115      (
116          S = black_bean,
117          E1 \== italian_sausage,
118          E2 \== fried_steak,
119          D \== pudding_parfait
120      );
121      (
122          S \== black_bean,
123          E1 = italian_sausage,
124          E2 \== fried_steak,
125          D \== pudding_parfait
126      );
127      (
128          S \== black_bean,
129          E1 \== italian_sausage,
130          E2 = fried_steak,
131          D \== pudding_parfait
132      );
133      (
134          S \== black_bean,
135          E1 \== italian_sausage,
136          E2 \== fried_steak,
137          D = pudding_parfait
138      );
139      (
140          S \== black_bean,
141          E1 \== italian_sausage,
142          E2 \== fried_steak,
143          D \== pudding_parfait
144      ).

```

```

145
146 /*A special test to denote rule 4.*/
147 rule4(S, D) :-
148     (
149         S = mushroom,
150         D \== cookies
151     );
152     (
153         S \== mushroom,
154         D = cookies
155     );
156     (
157         S \== mushroom,
158         D \== cookies
159     ).
160
161 /*A general definition of rule 5.*/
162 /*This rule is too general to apply in individual cases.*/
163 rules5(S, E, D) :-
164     (
165         S \== clam_chowder,
166         E \== italian_sausage,
167         D \== pudding_parfait
168     );
169     (
170         (
171             (S \== clam_chowder, E \== italian_sausage, D = pudding_parfait);
172             (S \== clam_chowder, E = italian_sausage, D \== pudding_parfait);
173             (S = clam_chowder, E \== italian_sausage, D \== pudding_parfait)
174         ),
175         servedMonday(clam_chowder, _, _),
176         servedTuesday(_, _, pudding_parfait),
177         servedWednesday(_, italian_sausage, _)
178     );
179     (
180         (
181             (S \== clam_chowder, E \== italian_sausage, D = pudding_parfait);
182             (S \== clam_chowder, E = italian_sausage, D \== pudding_parfait);
183             (S = clam_chowder, E \== italian_sausage, D \== pudding_parfait)
184         ),
185         servedTuesday(clam_chowder, _, _),
186         servedWednesday(_, _, pudding_parfait),
187         servedThursday(_, italian_sausage, _)
188     );
189     (
190         (
191             (S \== clam_chowder, E \== italian_sausage, D = pudding_parfait);
192             (S \== clam_chowder, E = italian_sausage, D \== pudding_parfait);
193             (S = clam_chowder, E \== italian_sausage, D \== pudding_parfait)
194         ),
195         servedWednesday(clam_chowder, _, _),
196         servedThursday(_, _, pudding_parfait),
197         servedFriday(_, italian_sausage, _)
198     ).
199
200 /*A special test to denote rule 6.*/
201 rules6(E, D) :-
202     (
203         E = chicken_tandoori,
204         D \== cupcakes
205     );
206     (
207         E \== chicken_tandoori,
208         D = cupcakes
209     );
210     (
211         E \== chicken_tandoori,
212         D \== cupcakes
213     ).
214
215 /*Rules Monday must always abide by.*/
216 ifMonday(S, E, D) :-

```

```

216 ifMonday(S, E, D) :-
217     isSoup(S),
218     isEntre(E),
219     isDessert(D),
220     S \== veggie_orzo,
221     S \== clam_chowder,
222     S \== split_pea,
223     E \== turkey_burger,
224     E \== italian_sausage,
225     D \== bundt_cake,
226     D \== pudding_parfait,
227     D == apple_pie,
228     rule6(E, D),
229     rule2(S, E),
230     rule3(S, E, E, D),
231     rule4(S, D).
232
233 /*Rules Tuesday must always abide by.*/
234 ifTuesday(S, E, D) :-
235     isSoup(S),
236     isEntre(E),
237     isDessert(D),
238     S \== split_pea,
239     E \== italian_sausage,
240     D \== apple_pie,
241     D \== bundt_cake,
242     D \== pudding_parfait,
243     rule6(E, D),
244     rule2(S, E),
245     rule3(S, E, E, D),
246     rule4(S, D).
247
248 /*Rules Wednesday must always abide by.*/
249 ifWednesday(S, E, D) :-
250     isSoup(S),
251     isEntre(E),
252     isDessert(D),
253     S \== black_bean,
254     S \== split_pea,
255     E \== fried_steak,
256     E \== italian_sausage,
257     E \== roasted_pork,
258     D \== apple_pie,
259     D \== bundt_cake,
260     D \== pudding_parfait,
261     rule6(E, D),
262     rule2(S, E),
263     rule3(S, E, E, D),
264     rule4(S, D).
265
266 /*Rules Thursday must always abide by.*/
267 ifThursday(S, E, D) :-
268     isSoup(S),
269     isEntre(E),
270     isDessert(D),
271     S \== clam_chowder,
272     S \== split_pea,
273     E \== roasted_pork,
274     D \== apple_pie,
275     rule6(E, D),
276     rule2(S, E),
277     rule3(S, E, E, D),
278     rule4(S, D).
279
280 /*Rules Friday must always abide by.*/
281 ifFriday(S, E, D) :-
282     isSoup(S),
283     isEntre(E),
284     isDessert(D),
285     S \== clam_chowder,
286     S == split_pea,
287     E \== roasted_pork,

```

```

288     D \== apple_pie,
289     D \== cookies,
290     D \== pudding_parfait,
291     rule6(E, D),
292     rule2(S, E),
293     rule3(S, E, E, D),
294     rule4(S, D).
295
296
297
298
299
300
301
302     /*=====*/
303     /*=====*/
304     /*||           Relations           ||*/
305     /*|| Rules dictating relations between days. ||*/
306     /*=====*/
307     /*=====*/
308
309
310
311     /*Declaring the rules for Monday*/
312     servedMonday(S, E, D) :-
313         isSoup(S),
314         isEntre(E),
315         isDessert(D),
316         S \== veggie_orzo,
317         S \== clam_chowder,
318         S \== split_pea,
319         E \== turkey_burger,
320         E \== italian_sausage,
321         D \== bundt_cake,
322         D \== pudding_parfait,
323         D == apple_pie,
324         rule6(E, D),
325         rule2(S, E),
326         rule3(S, E, E, D),
327         rule4(S, D),
328         /*rule 1 simplified for Monday*/
329         (
330             ( S \== clam_chowder, E \== roasted_pork, D \== bundt_cake );
331             (
332                 (S \== clam_chowder, E = roasted_pork, D \== bundt_cake),
333                 ifTuesday(clam_chowder, _, _),
334                 ifThursday(_, _, bundt_cake)
335             )
336         ),
337         /*rule 5 simplified for Monday*/
338         (
339             ( S \== clam_chowder, E \== italian_sausage, D \== pudding_parfait );
340             (
341                 (S = clam_chowder, E \== italian_sausage, D \== pudding_parfait),
342                 ifTuesday(_, _, pudding_parfait),
343                 ifWednesday(_, italian_sausage, _)
344             )
345         ).
346
347
348
349
350     /*Declaring the rules for Tuesday*/
351     servedTuesday(S, E, D) :-
352         isSoup(S),
353         isEntre(E),
354         isDessert(D),
355         S \== split_pea,
356         S \== mushroom,
357         E \== italian_sausage,
358         D \== apple_pie,
359         D \== bundt_cake,

```

```

360     D \== pudding_parfait,
361     rule6(E, D),
362     rule2(S, E),
363     rule3(S, E, E, D),
364     rule4(S, D),
365     /*rule 1 simplified for Tuesday*/
366     (
367         ( S \== clam_chowder, E \== roasted_pork, D \== bundt_cake );
368         (
369             (S = clam_chowder, E \== roasted_pork, D \== bundt_cake),
370             ifThursday(_, _, bundt_cake),
371             ifMonday(_, roasted_pork, _)
372         );
373         (
374             (S \== clam_chowder, E = roasted_pork, D \== bundt_cake),
375             ifWednesday(clam_chowder, _, _),
376             ifFriday(_, _, bundt_cake)
377         )
378     ),
379     /*rule 5 simplified for Tuesday*/
380     (
381         ( S \== clam_chowder, E \== italian_sausage, D \== pudding_parfait );
382         (
383             (S = clam_chowder, E \== italian_sausage, D \== pudding_parfait),
384             ifWednesday(_, _, pudding_parfait),
385             ifThursday(_, italian_sausage, _)
386         )
387     ).
388
389
390
391
392
393     /*Declaring the rules for Wednesday*/
394     servedWednesday(S, E, D) :-
395         isSoup(S),
396         isEntree(E),
397         isDessert(D),
398         S \== black_bean,
399         S \== split_pea,
400         E \== fried_steak,
401         E \== italian_sausage,
402         E \== roasted_pork,
403         D \== apple_pie,
404         D \== bundt_cake,
405         D \== pudding_parfait,
406         rule6(E, D),
407         rule2(S, E),
408         rule3(S, E, E, D),
409         rule4(S, D),
410         /*rule 1 simplified for Wednesday*/
411         (
412             ( S \== clam_chowder, E \== roasted_pork, D \== bundt_cake );
413             (
414                 (S = clam_chowder, E \== roasted_pork, D \== bundt_cake),
415                 ifFriday(_, _, bundt_cake),
416                 ifTuesday(_, roasted_pork, _)
417             )
418         ),
419         /*rule 5 simplified for Wednesday*/
420         (
421             ( S \== clam_chowder, E \== italian_sausage, D \== pudding_parfait );
422             (
423                 (S = clam_chowder, E \== italian_sausage, D \== pudding_parfait),
424                 ifFriday(_, italian_sausage, _),
425                 ifThursday(_, _, pudding_parfait)
426             );
427             (
428                 (S \== clam_chowder, E \== italian_sausage, D = pudding_parfait),
429                 ifTuesday(clam_chowder, _, _),
430                 ifThursday(_, italian_sausage, _)
431             )

```

```

432     ).
433
434
435
436
437
438 /*Declaring the rules for Thursday*/
439 servedThursday(S, E, D) :-
440     isSoup(S),
441     isEntre(E),
442     isDessert(D),
443     S \== clam_chowder,
444     S \== split_pea,
445     E \== roasted_pork,
446     D \== apple_pie,
447     rule6(E, D),
448     rule2(S, E),
449     rule3(S, E, E, D),
450     rule4(S, D),
451     /*rule 1 simplified for Thursday*/
452     (
453         ( S \== clam_chowder, E \== roasted_pork, D \== bundt_cake );
454         (
455             (S \== clam_chowder, E \== roasted_pork, D = bundt_cake),
456             ifTuesday(clam_chowder, _, _),
457             ifMonday(_, roasted_pork, _)
458         )
459     ),
460     /*rule 5 simplified for Thursday*/
461     (
462         ( S \== clam_chowder, E \== italian_sausage, D \== pudding_parfait );
463         (
464             (S \== clam_chowder, E = italian_sausage, D \== pudding_parfait),
465             ifTuesday(clam_chowder, _, _),
466             ifWednesday(_, _, pudding_parfait)
467         );
468         (
469             (S \== clam_chowder, E \== italian_sausage, D = pudding_parfait),
470             ifWednesday(clam_chowder, _, _),
471             ifFriday(_, italian_sausage, _)
472         )
473     ).
474
475
476
477
478
479 /*Declaring the rules for Friday*/
480 servedFriday(S, E, D) :-
481     isSoup(S),
482     isEntre(E),
483     isDessert(D),
484     S \== clam_chowder,
485     S = split_pea,
486     E \== roasted_pork,
487     D \== apple_pie,
488     D \== cookies,
489     D \== pudding_parfait,
490     rule6(E, D),
491     rule2(S, E),
492     rule3(S, E, E, D),
493     rule4(S, D),
494     /*rule 1 simplified for Friday*/
495     (
496         ( S \== clam_chowder, E \== roasted_pork, D \== bundt_cake );
497         (
498             (S \== clam_chowder, E \== roasted_pork, D = bundt_cake),
499             ifWednesday(clam_chowder, _, _),
500             ifTuesday(_, roasted_pork, _)
501         )
502     ),
503     /*rule 5 simplified for Friday*/

```

```

504 (
505   ( S \== clam_chowder, E \== italian_sausage, D \== pudding_parfait );
506   (
507     (S \== clam_chowder, E = italian_sausage, D \== pudding_parfait),
508     ifWednesday(clam_chowder, _, _),
509     ifThursday(_, _, pudding_parfait)
510   )
511 ).
512
513
514
515 /*=====*/
516 /*=====*/
517 /*|| Solver ||*/
518 /*|| A rule that will solve the system. ||*/
519 /*=====*/
520 /*=====*/
521
522 solve(A, B, C, D, E, F, G, H, I, J, K, L, M, N, O) :-
523   /*Testing uniqueness to limit possibilities*/
524   dif(A,B),
525   dif(A,C),
526   dif(A,D),
527   dif(A,E),
528   dif(B,C),
529   dif(B,D),
530   dif(B,E),
531   dif(C,D),
532   dif(C,E),
533   dif(D,E),
534   dif(F,G),
535   dif(F,H),
536   dif(F,I),
537   dif(F,J),
538   dif(G,H),
539   dif(G,I),
540   dif(G,J),
541   dif(H,I),
542   dif(H,J),
543   dif(I,J),
544   dif(K,L),
545   dif(K,M),
546   dif(K,N),
547   dif(K,O),
548   dif(L,M),
549   dif(L,N),
550   dif(L,O),
551   dif(M,N),
552   dif(M,O),
553   dif(N,O),
554   /*Testing fits for each meal*/
555   servedMonday(A, F, K),
556   servedTuesday(B, G, L),
557   servedWednesday(C, H, M),
558   servedThursday(D, I, N),
559   servedFriday(E, J, O),
560   /*Writing results*/
561   write_ln(''),
562   write_ln('Monday'),
563   write_ln(A),
564   write_ln(F),
565   write_ln(K),
566   write_ln(''),
567   write_ln('Tuesday'),
568   write_ln(B),
569   write_ln(G),
570   write_ln(L),
571   write_ln(''),
572   write_ln('Wednesday'),
573   write_ln(C),
574   write_ln(H),
575   write_ln(M),

```



```
576     write_ln(''),
577     write_ln('Thursday'),
578     write_ln(D),
579     write_ln(I),
580     write_ln(N),
581     write_ln(''),
582     write_ln('Friday'),
583     write_ln(E),
584     write_ln(J),
585     write_ln(O).
```