R. One, Two, Tree (1/5)

R-1

a. [[*ice cream*] *soda*]
b. [[*science fiction*] *writer*]
c. [[*customer service*] *representative*]
d. [*state* [*chess tournament*]]
e. [[*Mars Rover*] *landing*]
f. [*plastic* [*water cooler*]]
g. [[*typeface design*] *report*]

R-2

[[*country song*] [*platinum album*]]

R-3

Default answer: A drama about control freaks (i.e., freaks about control), performed during a space mission (i.e., a mission to space).

Many other answers are possible as long as each of the bracketings is correctly defined. Below are examples of correct answers for each bracketing:

For "[*control freak*]":
- a person who is obsessive about having things his way

For "[[*control freak*] *show*]":
- a show that is run by control freaks
- a show that contains control freaks (i.e., the actors are control freaks)
- a show that is designed for or intended for control freaks
- a display of behavior by a control freak

For "[*space mission*]":
- a mission into space

R. One, Two, Tree (2/5)

For "[[*space mission*] [[*control freak*] *show*]]":

- a control freak show that is broadcast to audiences on space missions
- a control freak show that is set on a space mission
- a control freak show that is about space missions
- a display of behavior by control freaks; the display is witnessed on a space mission

Some examples of incorrect answers:

- the show contains control freaks that are interested in space missions
- the show contains control freaks who run or are on a space mission

These are incorrect because they attach "*space mission*" to "*control freak*" instead of attaching it to "*control freak show*". For these answers to be correct, the bracketing would have to be “[[[space mission] [*control freak*]] *show*]".

R-4

[[[[*family* [*board game*]] [*togetherness effect*]] [*government study*]] *author*].

Although the following might also be defensible:
[[[[*family* [[*board game*] *togetherness*]] *effect*] [*government study*]] *author*]

Or even perhaps:
[[[[[*family* [*board game*]] *togetherness*] *effect*] [*government study*]] *author*]

R. One, Two, Tree (3/5)

R-5

$f(5) = 14$
$f(6) = 42$
$f(7) = 132$

There are $f(5)$ bracketings of *togetherness effect government study author* — whatever $f(5)$ turns out to be! Similarly there are $f(3)$ (= 2) bracketings of *family board game*. So you have to list $f(3) \cdot f(5)$ bracketings that split the 8-word sequence into 3 words + 5 words like this. But the full list for $f(8)$ must also consider other splits, such as 1 word + 7 words. The general principle is that

$$f(n) = \sum_{k=1}^{n-1} f(k) \cdot f(n-k) \text{ for any } n > 1$$

You can therefore compute each line in the table from the previous lines. By the way, the resulting sequence of numbers is called the Catalan numbers: you can look it up.

R-6

To get [[*big fluffy*] *pancake*], we'd need adjective + adjective = adjective. To get [[*samurai short*] *sword*], we'd need noun + adjective = adjective or noun + adjective = noun.

R. One, Two, Tree (4/5)

R-7

a. There are only 3 bracketings (fewer than *f*(4)=5 because the rules from R-6 are "missing"):

[*roasted* [*red* [*potato pancake*]]] - a roasted red pancake made of potatoes
[*roasted* [[*red potato*] *pancake*]] - a roasted pancake made of red potatoes
[[*roasted* [*red potato*]] *pancake*] - a pancake made of roasted red potatoes

Note that the 4th logical possibility, a red pancake made of roasted potatoes, is not consistent with this word order: you'd have to call it a *red roasted potato pancake*.

b. There are 7 bracketings (fewer than *f*(5)=14):

[[*crazy monkey*] [[*cheap cider*] *house*]] - the house of crazy monkeys serves cider that is cheap
[[*crazy monkey*] [*cheap* [*cider house*]]] - the house of crazy monkeys that serves cider is cheap
[*crazy* [*monkey* [[*cheap cider*] *house*]]] - the crazy house of monkeys serves cider that is cheap
[*crazy* [*monkey* [*cheap* [*cider house*]]]] - the crazy house of monkeys that serves cider is cheap
[*crazy* [[*monkey* [*cheap cider*]] *house*]] - the crazy house serves cheap cider that's for monkeys
[[[*crazy monkey*] [*cheap cider*]] *house*] - the house serves cheap cider that's for crazy monkeys
[[*crazy* [*monkey* [*cheap cider*]]] *house*] - the house serves crazy, cheap cider that's for monkeys

Again, there are logical possibilities that are not consistent with the word order, such as a house of monkeys that serves crazy, cheap cider.

R-8

a. 0
b. 1
c. 5. This generalizes R-7a. There are 5 ways to divide *Adj Adj Adj Adj* into an initial group that modifies the 2nd noun and a final group that modifies the 1st noun. Groups may be empty.
d. 14. First suppose that the nouns are bracketed as [[*Noun Noun*] *Noun*]. Then there are 10 ways to divide *Adj Adj Adj* into 3 groups which will respectively modify the 3rd, 2nd, and 1st noun (There are 4+3+2+1=10 ways to place two vertical dividers into this sequence. If the left divider falls before the first Adj, there are 4 positions for the right divider. If it falls before the second Adj, there are 3 positions for the right divider; and so on.) Alternatively, suppose that the nouns are bracketed as [*Noun*[*Noun Noun*]]. Then no adjective can modify the 2nd noun (it can only modify the whole [*Noun Noun*] compound), so then we divide *Adj Adj Adj* into only 2 groups as before; there are 4 ways to do this.

R. One, Two, Tree (5/5)

R-9

a. 1
b. 2
c. 3
d. 7
e. 12
f. 30

Let $g(n)$ be the number of bracketings of an alternating n-word sequence ending in *Noun*. Clearly, $g(1)=1$. To find $g(n)$, we can proceed as in R-5 and consider the ways of splitting the sequence into two shorter sequences, whose bracketings we count by applying g recursively. However, we have to leave out the splits where the first sequence ends in an adjective (i.e., where the second sequence has odd length), unless it's a single adjective (i.e., the first sequence has length 1). Thus, for odd $n>1$,

$$g(n) = g(n-2)g(2) + g(n-4)g(4) + \ldots + g(1)g(n-1),$$

while for even $n>1$,

$$g(n)= g(n-2)g(2)+g(n-4)g(4) + \ldots + g(2)g(n-2) +1\cdot g(n-1)$$

We can therefore compute

$$\begin{aligned}
g(2) &= g(1)g(1) &&= 1\\
g(3) &= g(1)g(2) &&= 1\\
g(4) &= g(2)g(2) + 1 \cdot g(3) &&= 2\\
g(5) &= g(3)g(2) + g(1)g(4) &&= 3\\
g(6) &= g(4)g(2) + g(2)g(4) + 1 \cdot g(5) &&= 7\\
g(7) &= g(5)g(2) + g(3)g(4) + g(1)g(6) &&= 12\\
g(8) &= g(6)g(2) + g(4)g(4) + g(2)g(6) + 1 \cdot g(7) &&= 30
\end{aligned}$$

Notice that these formulas are much faster than *listing* all the bracketings, which is important since $f(25)$ = 1,289,904,147,324 and even $g(25)$ = 50,067,108. The approach here can be generalized into a single powerful technique for rapidly counting the number of bracketings of *any* given word sequence.
Closely related algorithms are used to rapidly find the most likely bracketing or "parse" of a given sentence so that a computer can understand or translate it.