From the Nation's Leader in LSAT Preparation

LSAT LOGIC GAMES BIBLE

A Comprehensive System for Attacking the Logic Games Section of the LSAT

"Before reading the Logic Games Bible I was apprehensive about what I might encounter in the LSAT format—now I'm ready for anything. This has raised my level of comfort and confidence at least five-fold. I've shared some of the insights with other friends and they look at me, like 'Where did you learn to do that?' Thank you, PowerScore!"

Don Correia

POWERSCORE®

The following is a short excerpt from our *LSAT Logic Games Bible*, and is designed to illustrate PowerScore's methods and writing style.

Published by
PowerScore Publishing, a division of PowerScore Incorporated
57 Hasell Street
Charleston, SC 29401

Author: David M. Killoran

Manufactured in Canada
March 2010

ISBN: 978-0-9801782-0-3

Making Inferences

Inferences are relationships that must be true in a game but are not explicitly stated by the rules or game scenario. One of the keys to powerful games performance is making inferences after you have diagrammed all of the rules. In some games, a single inference can be the difference between the game seeming easy or difficult. For some people inference making is intuitive, and for others it is very difficult. Here are three basic but time-tested strategies for making inferences:

Linkage

Linkage should always be the first step you take to make inferences.

Linkage is the simplest and most basic way to make inferences. Linkage involves finding a variable that appears in at least two rules and then combining those two rules. Often that combination will produce an inference of value. Consider the following two rules:

> K must be played before L.
> L must be played before M.

Individually the two rules would be diagrammed as follows:

> K > L
> L > M

If we represented Not Laws from each rule, we would have the following:

1	2	3	4	5	6
~~L~~					~~K~~
~~M~~					~~L~~

Three variables in a sequence similar to the one to the right always yield 6 Not Laws.

Clearly, "L" is common to both rules. By combining the rules we come up with the following relationship:

> K > L > M

We can now infer that K must be played before M, and this information helps us to establish all of the applicable Not Laws:

1	2	3	4	5	6
~~M~~	~~M~~			~~K~~	~~K~~
~~L~~					~~L~~

Linkage between the rules should always be the first place you look to discover inferences. Incidentally, the above example again proves the value of reading all of the rules before you begin diagramming. If you had diagrammed each rule individually, then later discovered the linkage, you would then have to return to the two rules and diagram the additional implications of the linkage. That would be an inefficient approach and thus detract from your performance.

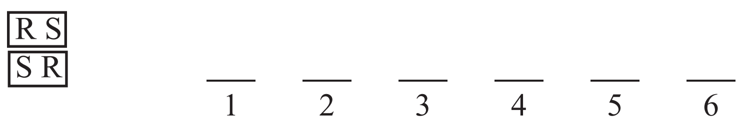
Here is another example featuring linkage:

These examples of linkage all relate to Linear games. Other linkage examples will appear in the chapters devoted to other game types.

> If Q is displayed fourth, then R must be displayed first.
> R and S are displayed consecutively.

Individually the two rules would be diagrammed as follows:

Q4 ⟶ R1



However, if R is first, then according to the second rule S must be second, which should be written as:

Q4 ⟶ R1, S2

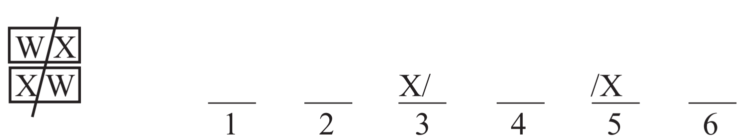
Thus, combining the two rules leads to the further inference that S must be second when Q is fourth. Although this cannot be represented directly on the diagram, this relationship can and should be displayed as above, as an addition to the original representation of the first rule.

Here is final example featuring linkage:

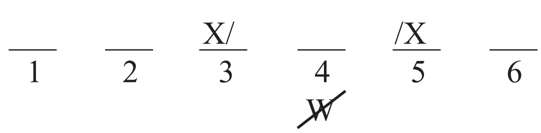
> W and X cannot speak consecutively.
> X must speak third or fifth.

Individually the two rules would be diagrammed as follows:

Although these three examples each feature linkage between two rules, there are situations that arise where three rules are linked.



At first, it may appear that linking the two rules yields no inference, but if X is always third or fifth, then W can never be placed fourth (to do so would cause a violation regardless of whether X was third or fifth). This leads to a W Not Law on 4:

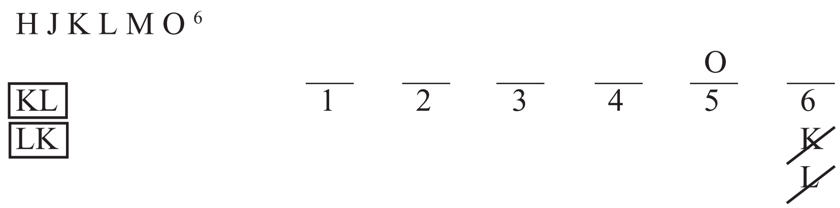


Rule Combinations

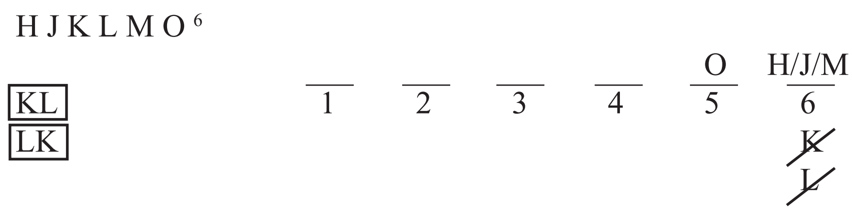
As we study more and more game types, your arsenal of rule recognition skills will increase. In certain games, there are classic combinations which always yield certain inferences. In contrast to linkage, however, making inference from rule combinations does not rely upon the connection of a variable common to two or more rules. For example, consider this scenario:

> Six lawyers—H, J, K, L, M, and O—must speak at a convention. The six speeches are delivered one at a time, consecutively, according to the following restrictions:
>
> K and L must speak consecutively.
>
> O must speak fifth.

From the scenario and rules above, we can draw the following diagram:



Because of the interaction of the two rules, we can infer that K and L can never speak sixth (there is not enough room for K and L to be next to each other). In addition, because O must speak fifth, only H, J, and M remain as possible candidates to speak sixth. This could be shown as a triple-option (H/J/M):



This type of rule combination is one of many we will discuss in this book.

Restrictions

In Logic Games always look to the restricted points for inferences. Restricted points are the areas in the game where only a few options exist—for example, a limited number of variables to fill in a slot, a block with a limited number of placement options, or a slot with a large number of Not Laws. If you can identify a restriction, generally there are inferences that will follow from your examination of that point. The trick is to determine exactly where the restrictions in a game actually occur.

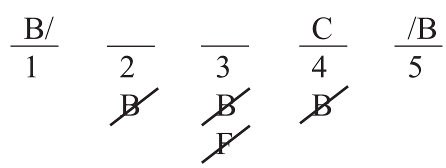
Note that rule combination inferences are normally produced by a restriction that results from combining the two rules.

Consider the following example:

> A salesman must visit five families—the Browns, the Chans, the Duartes, the Egohs, and the Feinsteins—one after another, not necessarily in that order. The visits must conform to the following restrictions:
>
> The Browns must be visited first or fifth.
> The Feinsteins cannot be visited third.
> The Chans must be visited fourth.

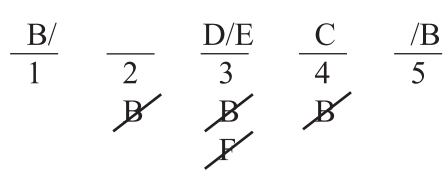
Using the scenario and rules above, we can produce the following diagram:

B C D E F ^{5}



The easiest way to find restrictions in a game is to examine the Not Laws for each slot. The slot with the most Not Laws may be so restricted that it has a limited number of possibilities. In this case, the third slot is the most restricted active slot since it has two Not Laws. Technically, the fourth slot is the most restricted since it has only one option, the Chans. But, since the fourth slot has already been filled by the Chans, it is no longer "active" and we can disregard it from further consideration. However, since the Chans have been placed, they cannot go in any other slot, and so it is now true that neither B, C, nor F can be visited third. Since there are only five families to visit and B, C, and F have been eliminated from contention, it follows that either D or E must be visited third. That inference should be shown with a D/E dual-option on the third slot:

B C D E F ^{5}



Restrictions also frequently occur with blocks, especially split-blocks. Consider the following scenario:

> A child must play five games—P, Q, R, S, and T—one after another, not necessarily in that order. The games must be played according to the following conditions:
>
> The child plays exactly two games between playing S and playing T, whether or not S is played before T.
>
> P is played immediately before Q is played.

Once again, using the scenario and rules, we can produce the following initial setup:

P Q R S T [5]

[S/T ___ ___ T/S]

[P Q]

___	___	___	___	___
1	2	3	4	5
~~Q~~		~~S~~		~~P~~
		~~T~~		

Blocks have a reduced number of spacing options and as such they can play a very powerful role in games.

As usual, P cannot be played fifth since Q must be played behind it, and Q cannot be played first since P must always be played ahead of it. However, the S and T split-block is more interesting because it has a limited number of spacing options. In fact, the ST split-block can only be placed into positions 1-4 or 2-5. Thus, S and T cannot be played third. At this point it appears that we have our inferences and that we are ready to continue on. But consider the interaction of the two blocks. If S and T are in the 1-4 position, then P and Q must be in 2-3. If S and T are in the 2-5 position, then P and Q must be in the 3-4 position. That means that either P or Q must always be played third. Additionally, we can infer that R must always be played first or fifth:

P Q R S T [5]

[S/T ___ ___ T/S]

[P Q]

R/	___	P/Q	___	/R
1	2	3	4	5
~~Q~~		~~S~~		~~P~~
		~~T~~		

Given these inferences, there are only two possible "templates" to the game based on the placement of the two blocks:

Template #2:	R	S/T	P	Q	T/S
Template #1:	S/T	P	Q	T/S	R
	1	2	3	4	5

These two templates encompass four solution sets to the game and make it abundantly clear how the interaction of some rules and game restrictions can set off a series of powerful inferences. In a later chapter we will discuss templates in detail, but you will see references to this approach throughout the book. Templates result from restrictions within the game, and in many games the best setup is one that shows these possibility blueprints.

Consider one more example:

> A doctor must see six patients—C, D, E, F, G and H—one after another, not necessarily in that order. The patients must be seen according to the following conditions:
>
> E is seen exactly three patients after C.
>
> D is seen immediately before F is seen.

Using the scenario and rules, we can produce the following initial setup:

C D E F G H ^{6}

C ___ ___ E

D F

___	___	___	___	___	___
1	2	3	4	5	6
~~E~~	~~E~~	~~E~~	~~C~~	~~C~~	~~C~~
~~F~~					~~D~~

Although this game has restrictions—the placement of the block is limited to 1-4, 2-5, or 3-6—there are still many different solutions. However, suppose for a moment that the test makers asked a question that contained a specific condition, such as:

> If G is seen third, which one of the following must be true?

The addition of this new condition affects the restrictions in the basic diagram to such an extent that only one solution is possible:

Step One: G is seen third

___	___	G	___	___	___
1	2	3	4	5	6

Step Two: The CE split-block must be placed into slots 1-4 because if it is placed in slots 2-5 there will be no room for the DF block

C	___	G	E	___	___
1	2	3	4	5	6

Step Three: The DF block must be placed into 5-6

C		G	E	D	F
1	2	3	4	5	6

Step Four: H must be placed into 2

C	H	G	E	D	F
1	2	3	4	5	6

Randoms are weaker variables that typically are placed after more powerful variables have been placed.

Note that the random H, a variable with little power, is placed last.

After completing these steps in response to the question, you could then use the single solution to easily select the correct answer choice.

The point is that restrictions, when present in a game, never go away. You must always track them and be prepared for questions that will force you to address the restriction.

Avoiding False Inferences

The test makers always check to see if you have interpreted the rules correctly, and some rules are easier to misinterpret than others. Here are four mistakes that students often make:

<u>Conditional Rule Reversal</u>

As previously discussed, a conditional rule is triggered when the sufficient condition occurs. For example, consider the following rule:

When M is shown first, then O is shown sixth.

This rule would be diagrammed as:

M1 ⟶ O6

When M appears in the first slot, then O *must* appear in the sixth slot. However, many test takers make the mistake of reversing the relationship, and when faced with O in the sixth slot, they assume that M must be in the first slot. This is not a valid inference, and this mistake is known as a Mistaken Reversal.

<u>Misinterpreting Block Language</u>

As discussed on pages 18 and 19, the test makers will use different language to denote different blocks. "Before" and "after" are used in one manner, whereas "between" has an entirely different meaning. Many students make the mistake of misreading this language, and then they draw inferences based on a relationship that is actually incorrect. You must always read each rule very carefully because making a mistake in the rules will almost always cause a high number of missed questions.

False Blocks

The test makers are savvy, and they know that the average student does not carefully read the language of the rules. Consider the following rule from a recent Linear game:

> Each rock classic is immediately preceded on the CD by a new composition.

In this game, songs were being selected for a demo CD, and each song was classified as either a rock classic (R) or a new composition (N). Most students, upon reading the rule above, immediately diagrammed the rule as follows:

N R

However, this representation is incorrect. The diagram above implies that R and N are always in a block formation, that is, every time N appears then R immediately follows, and every time R appears then N immediately precedes. Take a moment to re-read the rule. Does the rule state that the two variables are in block formation? No, what the rules states is that every rock classic is preceded by a new composition. There is no statement that every new composition is followed by a rock classic. So, this rule is only triggered when a rock classic is present. Thus, the rule is conditional, and should be diagrammed as follows:

R ⟶ [N R]

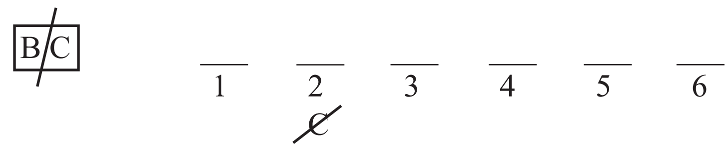
This representation correctly indicates that the relationship in the rule occurs when a rock classic is present. Under this representation, it becomes possible for two or more new compositions to appear in a row (NN, NNN, etc.).

False Not-Block Inferences

As discussed on page 21, not-blocks indicate that variables cannot be next to one another, or cannot be separated by a fixed amount of space. Some students make the mistake of combining not-blocks with Not Laws to arrive at false inferences. Consider the following two rules:

> B is not inspected the day before C is inspected.
> C cannot be inspected second.

The diagram for these two rules would be:



Many students, after reviewing these two rules, make one of the following two errors:

1. They mistakenly conclude that B cannot be inspected first, and then place a B Not Law under slot 1. These students erroneously act as if the BC relationship is a positive block (in a positive block, if C could not be inspected second, then B could not be inspected first).

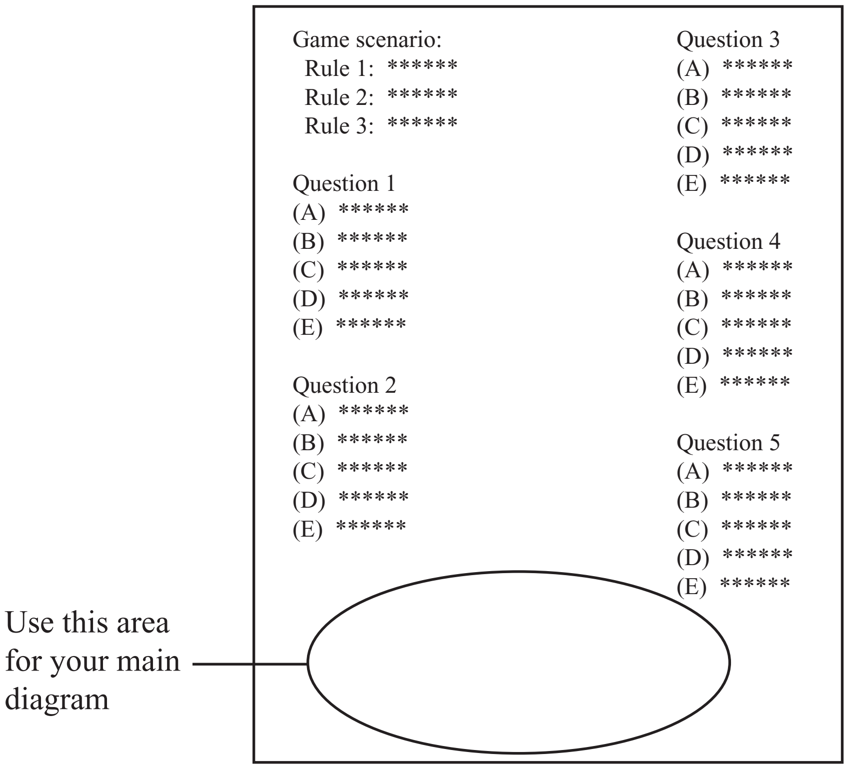
2. They mistakenly conclude that B must be inspected second, and then place B into slot 2. The error here is to act as if the C Not Law on slot 2 then forces B into slot 2. That outcome does not have to occur, although, of course, B could be inspected second.

Remember, not-blocks only come in to play when one of the variables in the not-block is placed on the diagram.

Also from PowerScore's LSAT Logic Games Bible:

Placing Diagrams

When you make your main diagram to represent all of the rules, you should create that diagram at the bottom of the page in the empty space beneath the questions, as indicated in the example below:



The sole exception to this rule is when the test makers supply a diagram within the game scenario and you decide to use that diagram.

When creating your main diagram, do not write too small or too large, do not crowd the rules together, and do not orient your diagram too far to the left or right. In some instances you may find that you need more room on one side than you previously thought, and it is helpful if you still have some room on each side of the diagram to work with. And, of course, you must write neatly.

Remember, you cannot use scratch paper during the scored portions of the LSAT, and so the space at the bottom of each game is the only space you will be able to use to make your main diagram. Yes, we know that there is not much space to work with on some games, but this is one of the challenges set by the test makers.

The process of creating hypotheticals is easier in practice than it sounds in theory, and this book often refers to hypothetical scenarios to explain why certain answers are right or wrong.

Diagramming Local Questions

Always do the work for Local questions next to the question itself.

For many Global questions, the work you do in creating your main setup and making inferences will be sufficient to answer the question. Local questions, which supply you with a new piece of information specific to that question only, generally require additional work. You should do this work next to the question itself. This affords you with several benefits:

1. By doing the work next to the problem, you reduce the visual disconnection between your work and the question. This saves you time.

2. If you need to come back to a question, when you return you will be able to see the work you did up to that point.

3. Should you be able to reuse the work you did for the question, you will be able to see the conditions that created that work more easily.

There are two alternate theories to this approach that are widely propagated and we believe each is flawed. Let us take a moment to examine why:

<u>Flawed Approach #1</u>: Do the work for each question on the main diagram

Do not erase any of your work unless you have made a mistake.

This approach suggests that the work for each question should be done on the main diagram. In order to use this method, you must erase your previous work before beginning each question. Erasing your work has a number of negative effects: you could accidentally erase important information that applies to all questions, you could accidentally leave information that applies only to one question, and most importantly, every time you erase your work you lose some of the knowledge that you created about the game. This method also destroys your ability to reuse your work, an approach discussed below. As a rule, *never* erase any of your work unless you have made a mistake.

<u>Flawed Approach #2</u>: Create a "grid" and do the work for each question in rows within the grid

This approach requires that you create a grid near the main setup. The work for each question is then done within rows of the grid, as follows:

	1	2	3	4	5	6
Question #1	T	V	W	Z	X	S
Question #2	V	W		Z		
Question #3		S	X	Z	T	

Although this method is superior to using the first flawed approach above, it too has several negative effects:

1. Drawing the grid and a main setup requires a large amount of space—space that the test makers do not always provide. In contrast, doing the work next to the question is space-efficient.

2. Using the grid creates a three-step visual disconnect: Your eye must travel between the question, the main setup, and the grid. This takes time and can cause confusion. In contrast, doing the work next to the question requires only a two-step visual disconnect: Your eye travels between the question and the corresponding work to the main setup. This saves time and eliminates confusion.

3. Using the grid tends to train students to use horizontal setups that do not contain a vertical component. As will be discussed in the Advanced Linear games section, this is particularly troublesome because the most complex Linear games have both vertical and horizontal components. For example, consider the following game. In this game music classes, three teachers, and three students were each assigned to one of four consecutive days. This produced a setup akin to the following:

Student:	___	___	___	___	(R, S, T)
Teacher:	___	___	___	___	(G, H, J)
Class:	___	___	___	___	(B, C, D, E)
	1	2	3	4	

   Trying to reproduce a similar diagram within a grid is a nightmare as each question requires three rows, something the grid is ill-equipped to handle. Any technique you use should work equally well for the hard questions and the easy questions. This is not the case for the grid.

4. As Linear games become more complex, the grid tends to work less and less efficiently. In contrast, working next to the question is always efficient since it allows you to draw the most appropriate diagram for the conditions.

By the way, it seems that many proponents of the grid also use notations such as "X," "O,", and "✓." This type of notation is relatively useless because it abstracts the representation process. Variables are always better represented by directly placing them on the diagram and using Not Laws, etc.

Any setup that use an X, O, or check is weak. See Appendix Three for more information.

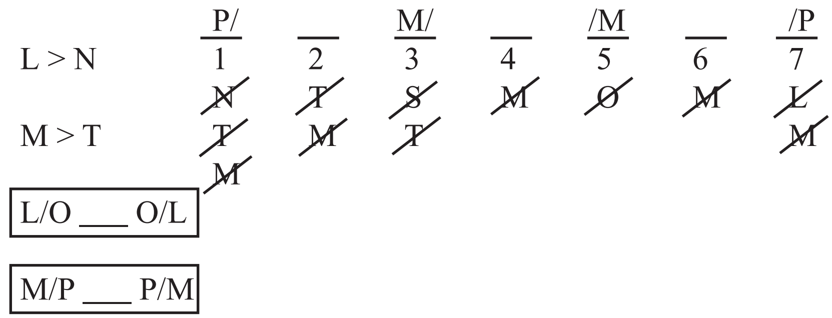
Now that we have discussed some flawed methods of diagramming the questions, let us return to the discussion of working next to the question itself. Working next to the question often requires you to re-create the basics of the main setup. This re-creation will be in skeletal form only; there is no need to redraw the entire diagram or all of the rules. Consider the following example:

The game setup to the right is provided in order to give you a sense of the difference between the main diagram and the mini-diagram you would create next to an individual question, and so the discussion that follows about "diagramming next to the question" makes sense. It is not intended to be an actual game diagramming exercise (if it had been, we would have written out the entire text of the game for you). Thus, not all the rules are included.

The key point on that page is that when work is done next to a question, only the bare basics of the diagram are re-created next to the question.

Main diagram from the second game of the September 1998 LSAT:

L M N O P S T [7]
*



Question #11 from the same game. Do the work next to the question!

11. If T is delivered fourth, the seventh package delivered must be

(A) L
(B) N
(C) O
(D) P
(E) S

Note that the work is done on a diagram that reflects just the simple base of the game—none of the rules are redrawn and you can even skip numbering the spaces if you feel comfortable.

Reusing Information

Reusing applicable information is one of the most powerful techniques available to students.

As you work with each question, your knowledge of the game naturally increases. As you approach each question, do not forget that you may have already gained information in a previous question that might apply to the question you are working on. Some students, upon hearing this advice, respond by saying, "But isn't the information in each question relevant to that question only?" Yes, the conditions in the question stem are relevant to that question only, but the work done for that question *may* apply elsewhere. Consider the following scenario:

> The first question of the game is a Global, Could Be True, List question, such as this one: