Solutions to Exercises in

Game Theory
Chapter 9

1. [Typo: The Duplicator has a winning strategy for the 2-round EF game, the Spoiler has a winning strategy for the 3-round game.] Consider the 2-round game: If the Spoiler chooses from A, say a_1 in the first round, the Duplicator must choose something from B, say b_1. If Spoiler chooses from B, then Duplicator chooses a_1. In the second round, if Spoiler chooses from A, say a_2, then Duplicator chooses a point adjacent to b_1, and if Spoiler chooses from B, then the choice is either b_1, in which case Duplicator chooses a_1 if this was Spoiler's choice in the first round, or if Spoiler also has chosen b_1 in the first round, otherwise Duplicator chooses a point in A different from a_1. Then f is a local isomorphism, so Duplicator has a win.

For the three round game, Spoiler chooses a_1, a_2, a_3. Then Duplicator must choose three different points of B which are all adjacent, and since this is not possible, the strategy is winning for Spoiler.

2. Clearly the graph $A_n = C_{2n}$ is Eulerian, having a cycle starting at b, moving to c via a_1, back to b via a_2, then to c via a_3 etc. and back to b via a_n. The graph $B_n = C_{2n+1}$ is not Eulerian, since an Euler cycle must use each of the points a_i in a movement in each direction, at the sum of these uses must be an even number.

Consider now the n-move EF game on A_n and B_n. Assume that Duplicator follows the following strategy: whenever the Spoiler chooses a point $b(c)$ in any of the graphs, then Duplicator chooses b (c) in the other graph. If Spoiler chooses a point a_i in any graph not used before, then Duplicator chooses the point a_j in the other graph with lowest index among the unused points, and if Spoiler chooses a point used before, the Duplicator uses the corresponding point in the other graph. It is seen that the result satisfies the equality and adjacency conditions and therefore $A_n \sim_n B_n$.

From Theorem 3 we may now conclude that the property of being Eulerian cannot be expressed in first order logic.

3. The simplest approach is by transforming Hackenbush to nimbers. The first case is a Nim with heaps of 3,2 and 1, and using Example 9.4, we find the its nimber from

	2^1	2^0
3	1	1
2	1	0
1	0	1
0	0	0

giving $*0$, so that Second mover wins. The second case is $*3 + *3 + *3 = *3$, so that First mover wins. Finally, in the third case, the middle tree ends with two edges giving $*0$, so that both can be deleted, taking us back to the first case, where Second mover wins.

4. Following the hint, we assign nimbers to towns. The town C has $\text{mex}(\{*0, *1, *2)$, D is downwards connected to C and A and has $\text{mex}(\{*1, *3\}) = *0$, and E has $\text{mex}(\{*0, *1\}) = *2$. The game corresponds to a Hackenbush with two trees having the nimbers $*1$ and $*2$, so that the game has number $*1 + *2 = *3$ (not $2 + *2$ as in the text), so that First mover wins.

5. (a) The game $\{-1 \mid 5\}$ is one where L has 1 independent moves (which does not change the position of R) and similarly R has 5 undependent moves. Clearly Second mover can enforce a win, so the game has value 0.

(b) The game $\left\{\frac{1}{4}\middle| 1\right\}$ can be viewed as a Blue-Red Hackenbush with two columns, the first one consisting of one blue followed upwords by two red, and a column consisting of one blue. Without changing the possibilities, Red may remove the top edge of the left column and Blue may remove the right column, leaving the game consisting of one blue followed upwards by one red, which corresponds to the game $\{0|1\} = \frac{1}{2}$.

6. The addition table can be constructed using mex as follows: We have $*n + *0 = *0$ for all n, and we use the recursive formula

$$*n + *m = \text{mex}(\{*n + *m' \mid m' < m\} \cup \{*n' + *m \mid n' < n\}).$$

Thus, the nimber $*1 + *1$ is found as $\text{mex}(\{*1 + *0\} \cup \{0 * +1*\}) = *0$, then $*1 + *2 = \text{mex}(\{*1 + *0, *1 + *1\} \cup \{*0 + 2*\}) = *3$, etc., giving the table

	1	2	3	4	5	6	7	8	9
1	0	3	2	5	4	7	6	9	8
2	3	0	1	6	7	4	5	10	11
3	2	1	0	7	6	5	4	11	10
4	5	6	7	0	1	2	3	12	13
5	4	7	6	1	0	3	2	13	12
6	7	4	5	2	3	0	1	14	15
7	6	5	4	3	2	1	0	15	14
8	9	10	11	12	13	14	15	0	1
9	8	11	10	13	12	15	14	1	0