Universität des Saarlandes
Fachrichtung 6.2 – Informatik
J. Siekmann, S. Autexier, K. Fischer, E. Melis, C. Zinn
M. Schiller, E. Schulz, R.Vollmann

2nd Theoretical Assignment in Artificial Intelligence (WS 2006/2007)
Solutions

Exercise 2.1 **(5 P)**
Explain P. Winston's (1980) claim:

> *More knowledge means less search.*

Can too much knowledge be problematic?

Solution:
In principle, when one has more knowledge, then one should be able to search in a more intelligent way (instead of blindly). In practice, with computer programs at least, often "more knowledge" means more options during search and actually degrades the performance of the search procedure.

Exercise 2.2 **(20 P)**
Cyclic graphs with marked nodes can be represented as follows: $edge(m, z, n)$ illustrates that an edge is drawn from node m to node n with the mark z (in the following, this notation describes a fact).

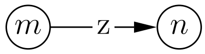


Figure 1: Example: edge from m to n with the mark z

1. Describe the following graph as a set of facts using this notation: (5 P)
   $1 \rightarrow^A 2 \rightarrow^B 3 \rightarrow^C 4 \rightarrow^B 5 \rightarrow^A 6$
2. Add these facts (i.e., the corresponding edges) to the following database: (10 P)

   axiom: $edge(m, z, n) \rightarrow palindrome(m, n)$
   $edge(m, z, m') \wedge edge(n', z, n) \wedge palindrome(m', n') \rightarrow palindrome(m, n)$

   theorem: $palindrome(1, 6)$

   Illustrate the search space as AND/OR tree to prove the theorem $palindrome(1, 6)$.
3. Which palindromes cannot be identified by this axiom? How could the database be extended to repair this fault. (5 P)

Hint: palindromes are words that result in the same word regardless if you read it forwards or backwards. For example, the graph $1 \rightarrow^A 2 \rightarrow^B 3 \rightarrow^C 4 \rightarrow^B 5 \rightarrow^A 6$ represents the word $ABCBA$.

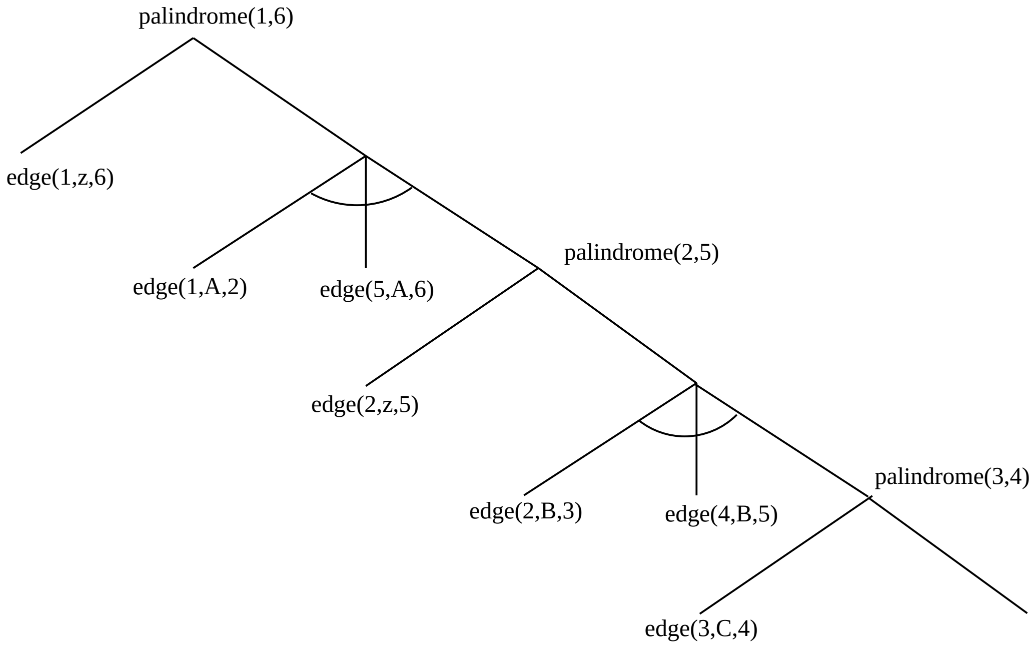
Solution:
For part 1, we convert the "graph notation" into the following "facts":
Facts:

- $edge(1, A, 2)$
- $edge(2, B, 3)$
- $edge(3, C, 4)$
- $edge(4, B, 5)$
- $edge(5, A, 6)$

For part 2, the "database" should be the facts above and the following two axioms:
Axioms:

- $edge(m, z, n) \rightarrow palindrome(m, n)$
- $edge(m, z, m') \wedge edge(n', z, n) \wedge palindrome(m', n') \rightarrow palindrome(m, n)$

The search space for the theorem $palindrome(1, 6)$ using the database is shown as an AND/OR tree below. The bottom node labelled $edge(3, C, 4)$ represents a successful termination of search since $edge(3, C, 4)$ is a fact in the database.



Part 3: Palindromes that could not be identified: palindromes of even size (size: number of letters). This could be fixed by adding an axiom $palindrome(m, m)$ where m is a variable.

Exercise 2.3 **(20 P)**

Examine the following version of the NIM-game:
Consider a heap with three and a heap with two matches. Two players, MAX and MIN, drag alternate one or more matches from **one** of the two heaps. MAX moves first. The player wins who has to move next and no more matches are available (vice versa: the player loses who has to take the last matches).

1. Illustrate the whole search space of this game. (5 P)
2. Evaluate the "win-situation" for MAX with +1 and MIN with -1. For all nodes, compute the values of the parent nodes. (5 P)
3. Can you identify a perfect strategy to win the game? If so, describe this strategy. If not, why? (5 P)

4. Assume, MAX can only project the results of two moves. Specify a function that leads to the same situation of the game. Name the disadvantages of this function. (5 P)

Solution:

Part 1: The search space is shown in Figure ??. Each node is associated with a search state. A search state is two number (n, m) where n is the size of the larger heap and m is the size of the smaller heap. (Note that after a turn which heap is "larger" may change.)

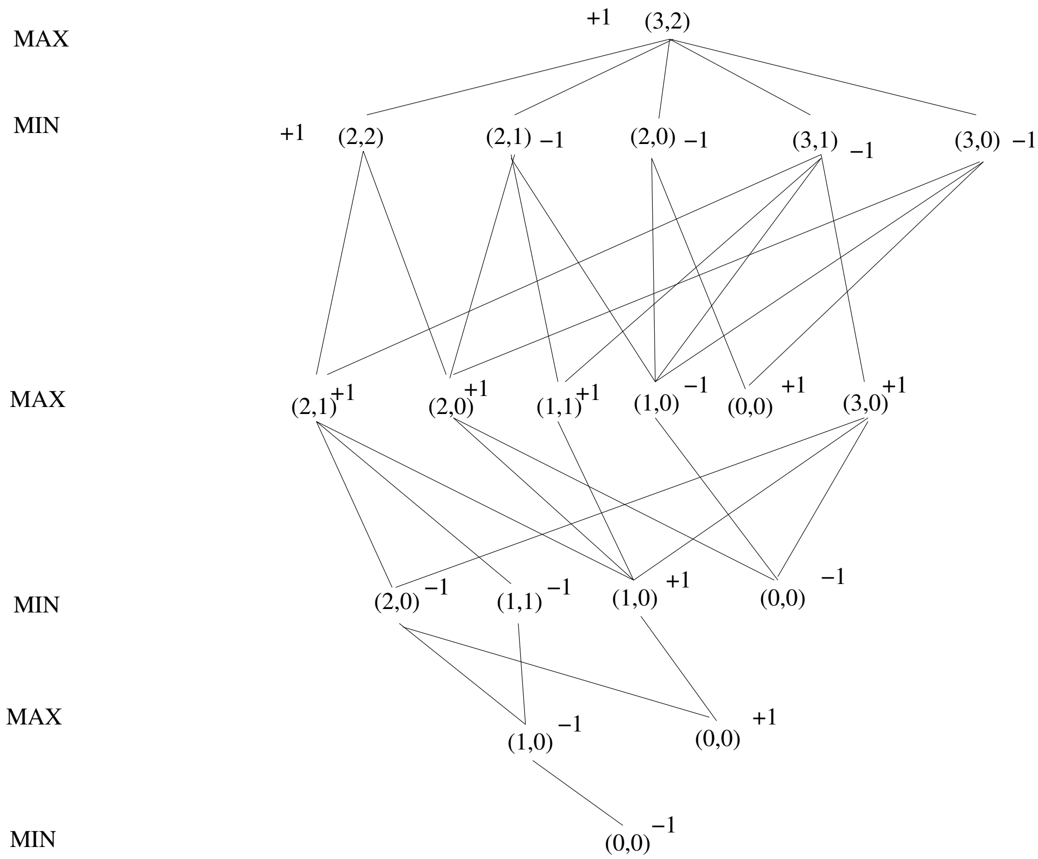


Figure 2: Search tree for NIM game

Note that there are alternative representations. For instance the state may be represented by (n, m) where n is the size of the "first" heap and m is the size of the "second" heap. Such a representation leads to redundancies when drawing the search space.

Part 2: In Figure ??, we included a +1 or -1 next to each node. First these are added to the four leaf nodes, then these are propagated to the parent nodes by taking MAX or MIN.

Part 3: A perfect strategy is for MAX to choose one match from the heap of three matches in the first move and then in the second move ensure that one heap has one match and the other heap is empty.

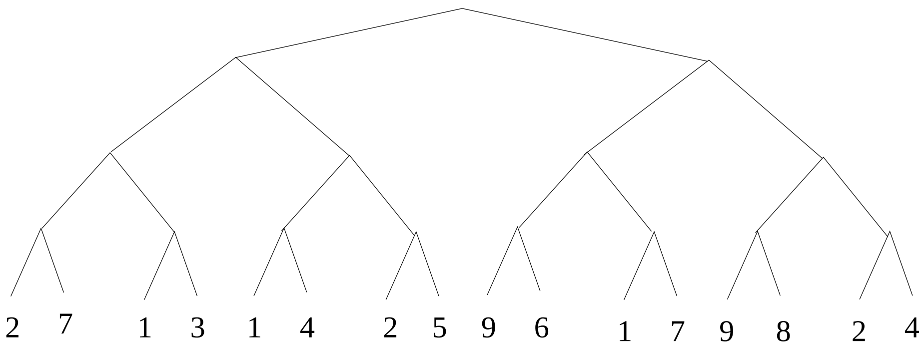
Part 4: Define the following function for tuples (a, b) in the state space:

$$f((a,b)) = \begin{cases} -1, & a = 1 \wedge b = 0 \\ +1, & \textit{otherwise} \end{cases}$$

Use the function f to evaluate the search state two steps ahead (after MAX and MIN have each taken one turn). Explanation: Note that f is evaluating a situation in which MAX has the next move. In the case of $a = 1 \wedge b = 0$, player MAX will lose the game as he/she has to take the last match. Otherwise, there still exists possibilities to win the game as player MAX is allowed to move next and thus can ensure the situation $a = 1 \wedge b = 0$ for MIN. (To see this,

consider each node in the search tree in Figure **??**.*) Examining the search space, if MAX uses a two step look ahead and f as an evaluation function, then the only possible first move for MAX is to take one match from the heap of three matches. After MIN moves, we are either in state $(2,1)$ or $(2,0)$. Two step look-ahead using f to evaluate leaves only the move to $(1,0)$ as a positive move. Note that two step look-ahead with this evaluation function f leads MAX to play the perfect strategy from part 3.*
Disadvantage: f evaluates the game perfectly, there is no disadvantage to using f. Perhaps asking for "disadvantages" of the function f was a tricky question? In general, an evaluation function might not evaluate a game perfectly.

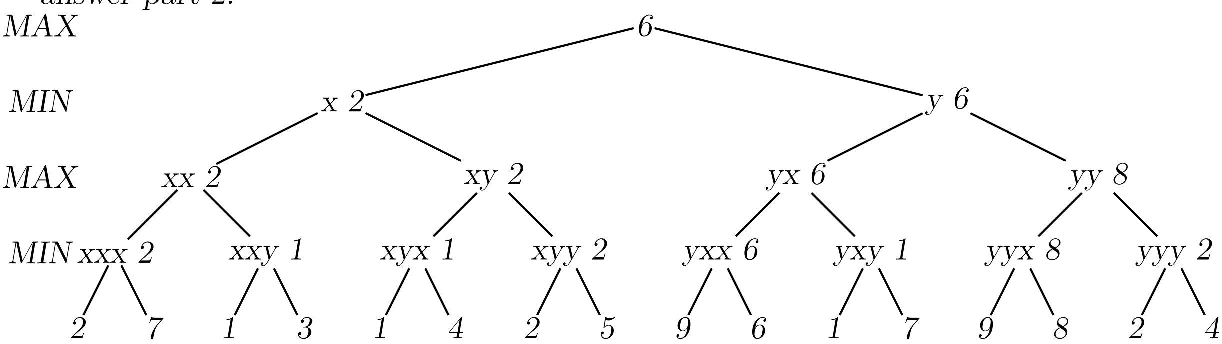
Exercise 2.4 **(10 P)**
MAX and MIN are playing a game. Assume that MAX is allowed to move next. The projection of the next four moves results in the following evaluated search space:



1. Identify the parent value for each node. Which move does MAX select next? Which leaf node does MAX expect to reach? (5 P)
2. Apply the alpha-beta strategy to this tree. In which order are the values evaluated? Which parts of the tree can be deleted? (5 P)

Solution:

Part 1: *Values are added to the parent nodes below. I also added labels to the nodes so I can answer part 2.*



Part 2: *Applying the alpha-beta strategy, the nodes/values are evaluated as follows:*

- *First we evaluate the left-most leaf to get 2.*
- *The backup value of xxx is now 2. That is, xxx will be at most 2.*
- *We evaluate the other leaf child of xxx to get 7.*
- *Taking the min, the value of xxx is 2.*

- *The backup value of xx is now 2. The value of xx will be at least 2.*
- *The left child of xxy is evaluated to obtain 1.*
- *The backup value of xxy is now 1. The value of xxy will be at most 1.*
- *Since 1 is less than the backup value of 2 at node xxy, the value of the node xxy will not change the value of xx.* **The right child of xxy can be deleted (pruned).** *Hence the value of xx is 2.*
- *The backup value of x is now 2. The value of 2 will be at most 2.*
- *The left child of xyx is evaluated to obtain 1 and the node xyx is given the backup value of 1.*
- *The right child of xyx is evaluated to obtain 4.*
- *The value of the node xyx is now 1 (the min of 1 and 4).*
- *The node xy now has backup value 1. The value of xy will be at least 1.*
- *The left child of xyy is evaluated to give 2.*
- *The node xyy is given backup value 2. The value of xyy will be at most 2. Since 2 is bigger than 1 (the backup value of xy), we continue to explore.*
- *The right child of xyy is evaluated to give 5.*
- *The value of xyy is 2 (the min of 2 and 5).*
- *The value of xy is 2 (the max of 1 and 2).*
- *The value of x is 2 (the min of 2 and 2).*
- *The root node is given backup value 2. The value of the root node will be at least 2.*
- *The left child of yxx is evaluated to give 9.*
- *The backup value of yxx is set to 9.*
- *The right child of yxx is evaluated to give 6.*
- *The value of yxx is 6 (the min of 9 and 6).*
- *The node yx is given backup value 6. The value of yx will be at least 6.*
- *The left child of yxy is evaluated to give 1.*
- *The node yxy is given backup value 1. The value of yxy will be at most 1 Since 1 is less than 6, the value of yx must be 6 and we can prune this part of the tree.* **The right child of yxy can be deleted.**
- *The value of yx is 6.*
- *The backup value of y is 6. The value of y will be at most 6.*
- *The left child of yyx is evaluated giving 9.*
- *The backup value of yyx is set to 9.*
- *The right child of yyx is evaluated giving 8.*
- *The value of yyx is 8 (the min of 9 and 8).*
- *The backup value of yy is 8. The value of yy will be at least 8.* **Since the value of yy will be at least 8 and the value of y will be at most 6,** $\alpha\beta$ **stops evaluating.**
- *The value of y must be 6.*
- *The value of the root node is 6 (the max of 2 and 6).*

Exercise 2.5 **(25 P)**

Consider the following problem:

> *Three missionaries and three cannibals are all together on one side of a river. There is one boat that can carry either one or two persons. How can they all cross the river such that there are never less missionaries than cannibals on either side of the river?*

1. How would you represent a state of that search problem? (10 P)
2. Which abstraction did you use in order to obtain an adequate representation? (10 P)
3. Draw the complete search tree for that problem. (Hint: Do not expand further the tree for states that occur for the second time on one path of the tree.) (5 P)

Solution:

1. **State** *A state consists of a triple* (m, k, b). *The first to components of the triple should be elements of* $\{1, 2, 3\}$. *The last component should be an element of* $\{s, z\}$. m *stands for the number of missionaries being at the "start" waterside and* k *for the number of cannibals at the "start" waterside.* b *stands for the waterside the boat is currently situated (*s *for start und* z *for goal).*

   Operations *A state changes if at least a single person boards in order to go to the next waterside. As one or two persons can board, we can distinguish between five different operations:*

   - *Two missionaries board and cross the river.*
   - *One missionary boards and crosses the river.*
   - *One missionary and one cannibal board ...*
   - *One cannibal boards ...*
   - *Two cannibals board ...*

   Goal-State *State* $(0, 0, z)$

   Costs *Number of times crossing the river.*
2. *The different missionaries and cannibals are illustrated using the same representation. In particular, we abstract away any distinguishing characteristics of the missionaries (such as their names) and any distinguishing characteristics of the cannibals (such as preference for white or dark meat). If we distinguished between different missionaries and cannibals, there would be far more states and operators. For example, instead of 2 missionaries on the start side and 1 on the goal side, we might have John and Paul on the start side with Monty on the goal side or we might have John and Monty on the start side with Paul on the goal side. Also, only the critical moments are considered, i.e., moments in which the boat lands on one waterside and the cannibals could attack the missionaries.*
3. *The search space is illustrated in Figure* **??**. *The start-state is the root of the search tree. The next possible states are illustrated on the next level and connected with the parent state. States in which the cannibals could attack the missionaries are illustrated as ovals. The search on a path can be stopped if a state is reached which was already explored. These states are illustrated as boxes.*

   Overall, there exists four feasible solutions (without repeating states). The node corresponding to the goal state is the $(0, 0, z)$ *node at the bottom.*

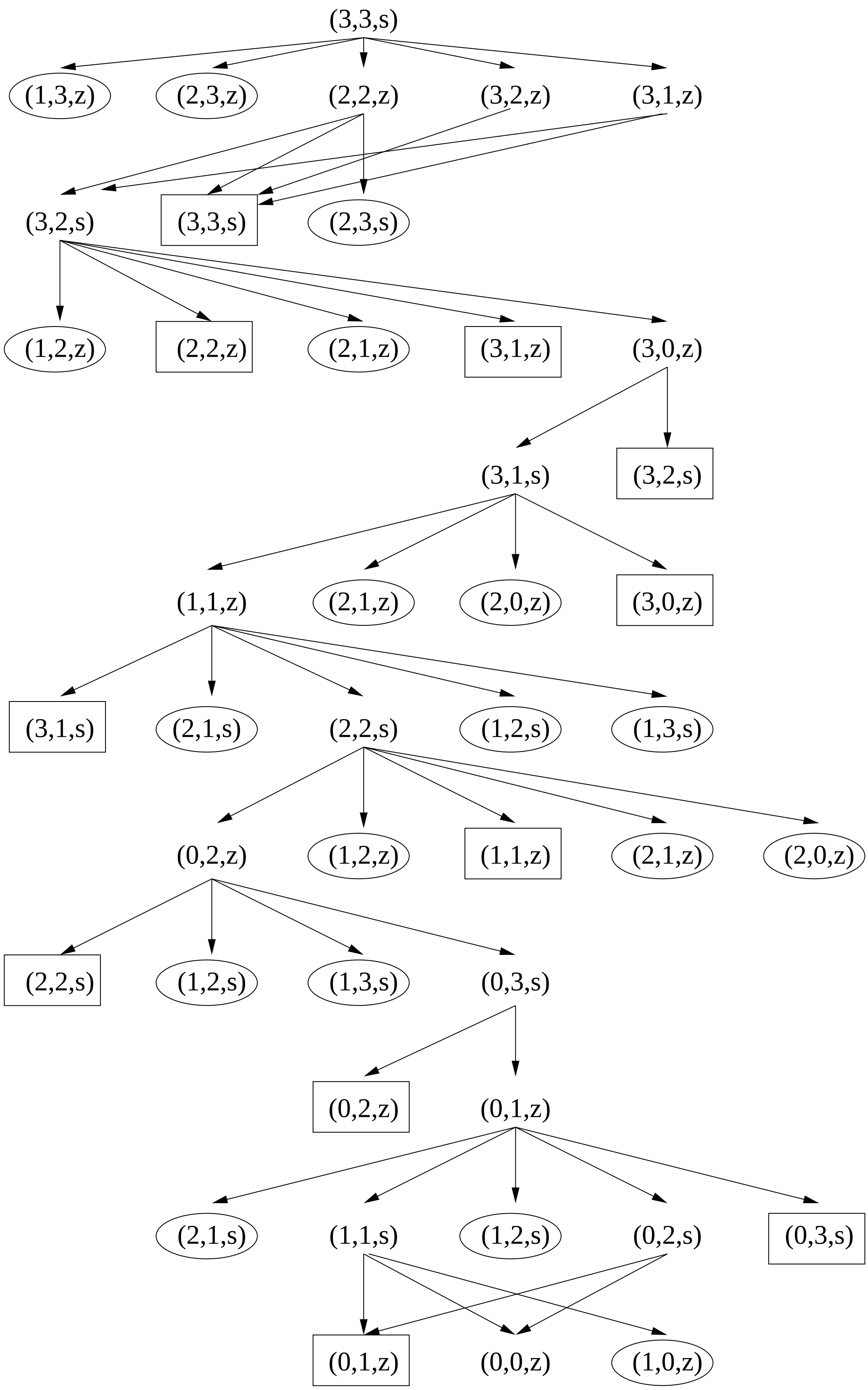


Figure 3: Search tree for the missionaries-cannibals problem

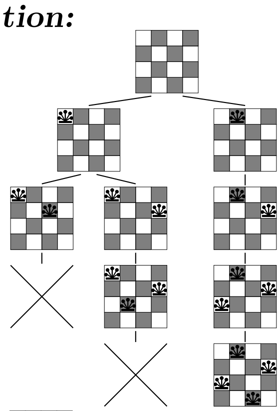
Exercise 2.6 **(20 P)**

Consider the 4-queens problem, where four queens are positioned on a 4×4-chess board such that no queen threatens any other queen.

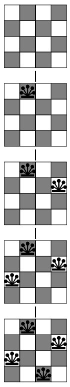
1. A procedure positions the queens row by row in the leftmost unthreatened column. The procedure performs backtracking; whenever a state is reached with no possibilities left to continue, then the search continues from the last choice point with unexplored ("next-leftmost") alternatives.
   Which states of the search space are generated until a solution is reached? (10 P)
2. The procedure is enhanced by the following heuristics: Let $D(i,j)$ be the length of the larger diagonal for the position (i,j). In every row i, the column j is chosen, for which $D(i,j)$ is minimal. A little example will explain the concept of the larger diagonal. The position $(1,3)$ has the two following diagonals: D_1 goes from $(1,3)$ to $(3,1)$, so it has a length of 3 (3 fields traversed) and D_2 goes from $(1,3)$ to $(2,4)$, so it has a length of 2 (2 fields traversed).
   Which states of the search space are generated until a solution is reached? (10 P)

Solution:

1.



2.



Note that the second search space does not backtrack. The heuristic was only used in the first step where the queen was put in position $(1,2)$ instead of $(1,1)$ since $D(1,1) = 4$ and $D(1,2) = 3$. For every subsequent step there was only one position on the next row which was not threatened.
