SAARLAND UNIVERSITY ARTIFICIAL INTELLIGENCE
Dr. Álvaro Torralba, Prof. Wolfgang Wahlster
Dr. Cosmina Croitoru, Daniel Gnad, Marcel Steinmetz
Yannick Körber, Michael Barz
Christian Bohnenberger, Sören Bund-Becker, Sophian Guidara,
Alexander Rath, Khansa Rekik, Julia Wichlacz, Anna Wilhelm

Exercise Sheet 2.
Solutions due Tuesday, **May 15**, 16:00 – 16:15, in the lecture hall.[1]

Exercise 5: Suboptimal A^* solution. (2 Points)

Construct an example where the A^* algorithm as defined in the lecture returns a suboptimal solution when executed with an **admissible, but inconsistent, heuristic**. Draw the relevant part of a search space and **annotate each node with its heuristic value, its g-value, as well as the order in which it has been selected for expansion.**

Hint: in the "h-weighted state space", as defined on slide 32 of Chapter 5, there must exist a transition with negative cost.

[1] Solutions in paper form only, and solution submission only at the stated time at the stated place. At most 3 authors per solution. All authors must be in the same tutorial group. All sheets of your solution must be stapled together. At the top of the first sheet, you must write the names of the authors and the name of your tutor. Your solution must be placed into the correct box for your tutorial group. If you don't comply with these rules, 3 points will be subtracted from your score for this sheet.

Exercise 6: Minimax. (2 Points)

Minnie and Macksy are playing a variant of the game Connect Three (*Drei gewinnt*) in which players take turns dropping colored discs down a 3×3 grid. Macksy's discs are yellow while Minnie's discs are red. The player chooses a column and the disc is placed at the first empty position starting from the bottom.

In their variant of the game, **Minnie wins with utility -1 if *any* line (horizontal, vertical or diagonal) fills up with three discs of the same color**, whereas **Macksy wins with utility $+1$ if the grid is full and does not contain such a line**.

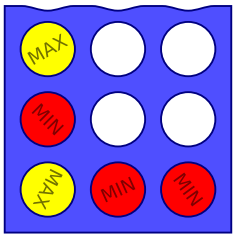


Figure 1: A 3×3 Connect Three grid. Macksy plays next.

(a) Consider the initial state depicted in Figure 1. Here, it is Macksy's turn to play.

Draw the full Minimax tree and annotate every node with its utility.

(b) Consider the evaluation function $f(x) :=$ the number of rows, columns and diagonals which contain at most one disc of the same color (do not contain two discs of the same color). For example, in the initial state $f(x) = 6$.

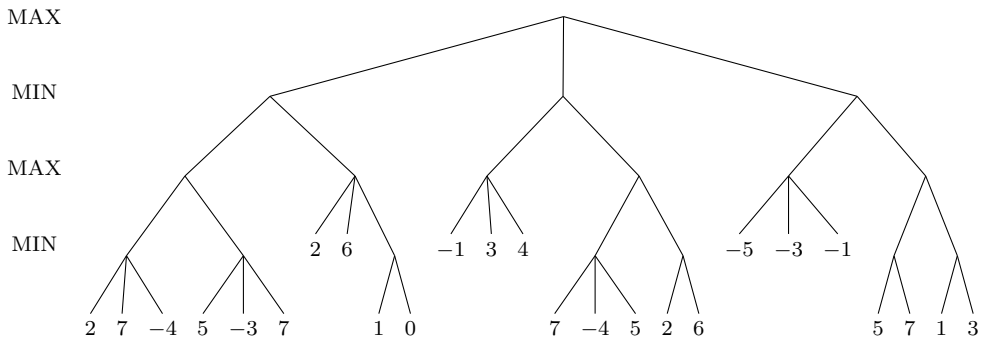
Note: The evaluation function is equivalent to 8 minus the number of rows, columns, and diagonals that *do* contain at least two discs of the same color.

Draw the **Minimax tree with this evaluation function and a depth of 2. Annotate every node with its utility.** Is this a good evaluation function? Justify your answer.

Exercise 7: Alpha-Beta pruning. (4 Points)

Consider the following game tree corresponding to a two-player zero-sum game as specified in the lecture. As usual, **Max is to start in the initial state (i.e., the root of the tree)**. For the following algorithms, the expansion order is **from left to right, i.e., in each node the left-most branch is expanded first.**

Note: See Figure 2 for a larger printable version of this tree.



(a) Perform Minimax search, i.e., annotate all internal nodes with the correct Minimax value.

(b) Perform Alpha-Beta search. For this, **annotate "all" internal nodes with the value that will be propagated to the parent node as well as the final** $[\alpha, \beta]$ **window before propagating the value to the parent (similar to slides 36 and 37 in Chapter 6).**

Mark which edges will be pruned. How many leaf nodes are pruned?

(c) There are move orderings that yield more effective prunings (as mentioned in slide 38, e.g., compare slides 37 and 40 in Chapter 6). Reorder the nodes of the tree such that **at least 16 leaf nodes are pruned**. A node counts as pruned in this context if it is not expanded or evaluated. **Perform Alpha-Beta search** on this tree as in the previous exercise.

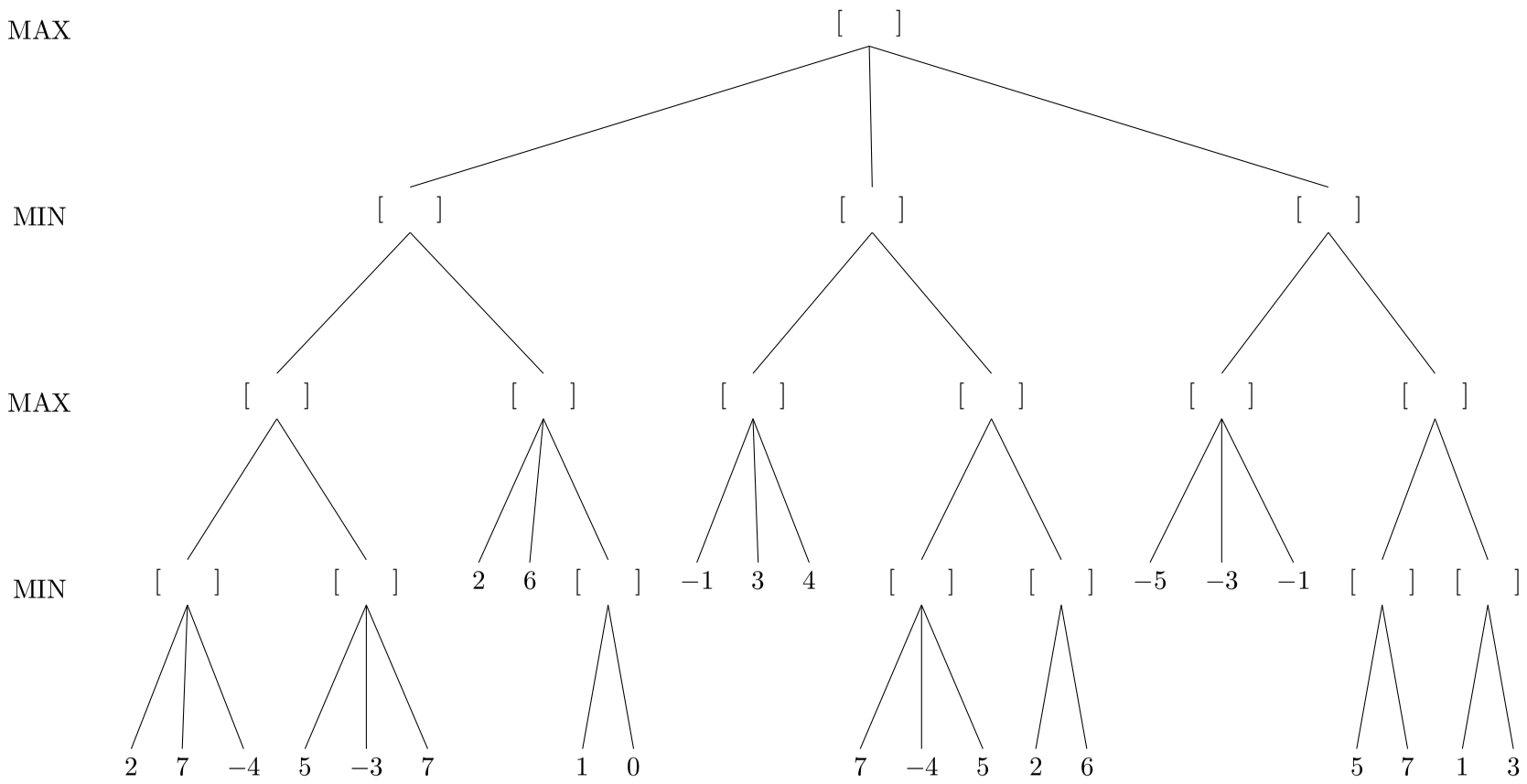
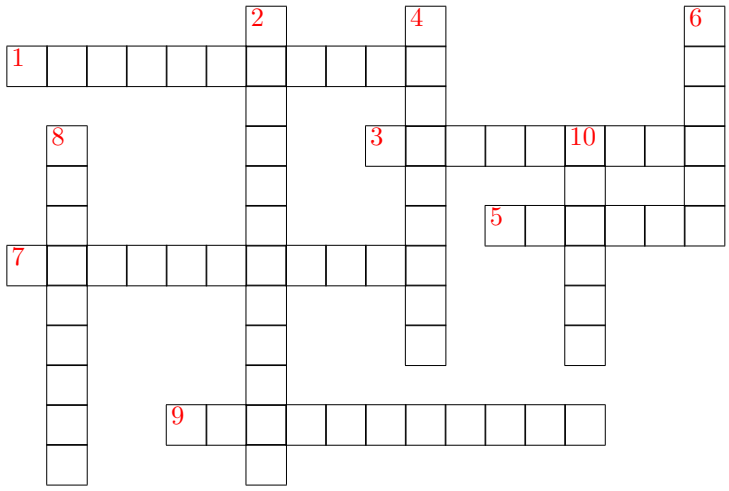


Figure 2: A zero-sum game tree.

Exercise 8: Formulating a constraint network. (2 Points)

Consider the crossword puzzle in Figure 3. Even numbers stand for words to be inserted vertically, odd numbers stand for words to be inserted horizontally. Each word can be added only once.




```
algorithm     inference
backtracking  information
computation   search
constraints   spaces
heuristic     states
```

Figure 3: Crossword puzzle to be formalized in Exercise 9.

Formulate this puzzle as a **constraint network whose solutions are the solutions to the puzzle.** Use the set of variables $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$, where variable v_i models which words can be put into gap i. The domain of each variable is a subset of the words to be inserted into the puzzle.

(a) Specify the domain of each variable, i.e., give D_{v_i}, for each variable $v_i \in V$.

(b) Specify the constraints, i.e., give $C_{v_i v_j} \subseteq D_{v_i} \times D_{v_j}$, for all relevant pairs of variables. Do not include the pairs where the set of feasible values is equal to the entire set $D_{v_i} \times D_{v_j}$.

Exercise 9: Naïve backtracking algorithm. (2 Points)

As the new CEO for the Svalbard Broadcasting Corporation, you decide to launch a local news radio station in each of the eight settlements on the territory (See the map on Figure 4). Since your broadcasting towers overlap with each other, you must use multiple frequencies to avoid interference between radio broadcasts. Unfortunately, your budget only allows you to purchase the rights to three different frequencies.

You decide to model your situation as a constraint satisfaction problem $\gamma = (V, D, C)$. Each tower has a variable $v \in V$ where

$$V = \{\text{Barentsburg, Bölscheøya, Hopen, Isbjørnhamna, Longyearbyen,}\\ \text{Ny-Ålesund, Piramida, Sveagruva}\}$$

and every v has domain $D = \{89.9\,\text{MHz}, 94.7\,\text{MHz}, 107.5\,\text{MHz}\}$.

The constraints are such that no pair of towers with overlapping range can broadcast on the same frequency, i.e., $C_{\{u,v\}} \in C$ for all overlapping towers u and v, $u \neq v$.

For example, $C_{\{\text{Bölscheøya,Hopen}\}} \in C$ because the Bölscheøya and Hopen towers have overlapping range. Conversely, $C_{\{\text{Barentsburg,Hopen}\}} \notin C$.

(a) Run the naïve backtracking algorithm using the combination strategy described on slide 42 of Chapter 8, i.e., **among the most *constrained* variables, pick the most *constraining* first. As a tie-breaker, use the alphabetical order on the region labels**. The value assignment order should be 89.9 MHz → 94.7 MHz → 107.5 MHz.

Draw the search graph and mark inconsistent assignments.

Note: Use the abbreviations "Ba," "Bö," "Ho," "Is," "Lo," "NÅ," "Pi," and "Sv" instead of the full names.

(b) When using the same *value* ordering (89.9 MHz → 94.7 MHz → 107.5 MHz), define a *variable* ordering such that the search graph contains more nodes than the one in the previous part. Explain why more nodes are generated.

Note: Do not draw the search graph. A simple explanation is enough.

Figure 4: A map of your eight local news broadcasting towers on Svalbard.