HOCHSCHULE RHEIN-WAAL RHINE-WAAL UNIVERSITY OF APPLIED SCIENCES FACULTY OF SOCIETY AND ECONOMICS

AUTOMATED SCENARIO BUNDLE ANALYSIS THROUGH ARTIFICAL INTELLIGENCE: USE OF CHATGPT

WRITTEN PAPER

BY

BENNET KUHLEN

JOHANNES RAUSCHENBERGER

TOM HAHN

HOCHSCHULE RHEIN-WAAL
RHINE-WAAL UNIVERITY OF APPLIED SIENCES
FACULTY OF SOCIETY AND ECONOMICS

AUTOMATED SCENARIO BUNDLE ANALYSIS THROUGH AI

A Written Report Submitted in
Partial Fulfilment of the
Requirements of the Degree of

Bachelor of Arts
In
INTERNATIONAL BUSINESS & MANAGEMENT

By
Johannes Rauschenberger, 33060
Bennet Kuhlen, 32301
Tom Hahn, 31891

Supervised by
Prof. Dr. Pitz

Submission Date
31-8-2024

Table of Contents

List of Figures

1 Introduction

Game theory is a recognized science that enables us to rationally depict, analyse and also evaluate interactions between individuals, companies, countries, etc. (in game theory all grouped under the collective term “players”). This applies both to past decisions (“strategies”) and to developments and interactions in the future. Like any science, game theory can sometimes become highly theoretical, and it has every right to do so, but we should not lose sight of the actual benefits of game theory. This “real use”, in the eyes of the authors of this paper, is to use game theory as a supportive basis for making decisions. Until recently, however, there have been enormous difficulties and obstacles that have made it impossible for game theory to realise its true potential. These obstacles are or were

a) the complexity of real-life situations

b) Apparently high mathematical complexity

c) Technological limitations

d) Underestimated relevance in practice

Everyone is aware of the complexity of reality. There are always countless variables, highly complex dependency structures and many players. It seems impossible to depict a real scenario in a scientific model. Any model seems too small and too imprecise for that.
To anticipate this, a scientific and more specific, game-theoretical model does not have to include every little variable to be useful and accurate, because it refers to the essential aspects of a problem. These essential aspects are also the ones that mainly influence decisions and outcomes in reality. Artificial intelligence is nevertheless able to analyse complex situations better than any algorithm before it. AI can analyse huge amounts of data and discover patterns and correlations that are difficult for humans to recognize. It can identify important variables and use these findings to refine models without explicitly integrating all the data into the model. AI could run mathematical theories in the background and reduce the mathematical requirements for use to a minimum. Otherwise, it would be possible to explain mathematical concepts in understandable language or to visualise the output. A major technological limitation for game theory was the lack of LLMs in connection with their increased computing power.
Regarding point d), the authors of this paper assume that with the increasing use of game theory, its relevance grows synchronously. In non-cooperative “games” or situations, game theory could become an advantage for the player who makes use of it.
Artificial intelligence could make game theory accessible and usable for the masses. This applies to everything from everyday life to life-changing decisions. Especially when it comes to major decisions, it is very likely that many people would be grateful for a science that does not take the decision away from them, but can help them make it.
How well artificial intelligence can help an individual person make a decision depends on how well the AI knows the person. The more the AI knows about a person and their situation, emotions, goals, etc., the more accurate the answers will be. This means that the output of the AI always correlates strongly with the input that was provided.
Every large language model (LLM) and every artificial intelligence depends on data that needs to be analysed and evaluated. An LLM has incredibly large amounts of data stored and this is particularly evident when it is asked to analyse conflicts or scenarios that are known to the AI and in which players are involved that it knows exactly.

The authors of this paper believe that current challenges in game theory, combined with the possibilities that artificial intelligence brings, could give a boost to Scenario Bundle Analysis, a rather unknown concept in game theory. Artificial intelligence could completely refresh scenario bundle analysis. Scenario bundle analysis, which will be discussed in more detail in the next chapter, is a game theory model that makes game theory more tangible and, above all, more useful for society.

2 Scenario Bundle Analysis

The scenario bundle analysis/ method is part of cooperative game theory and is an extensive game. Scenario Bundle Analysis is a method within Game Theory that was developed by Prof. Dr. Reinhard Selten. He presented "The Scenario Bundle Method" at a Research Conference on Strategic Decision Analyses Focusing on the Persian Gulf in 1977.

Scenario Bundle Analysis is used to examine (primarily geopolitical) conflicts, to compare the strategies each participant has available and to analyse the scenarios that result from the decisions they make.

Initially, a list is made comprising all relevant actors involved in the selected conflict. An actor may not have any internal conflict, but must rather be considered homogenous for the purposes of the analysis.
In the next step the attributes of each actor are listed and ranked. The ranking consists of natural numbers to ensure clear comparability between actors who share attributes.
Subsequent to the attributes, the actors' attitudes and intentions are compared. This includes aims and fears of certain outcomes that actors may harbour.
They are in turn classified on a numerical scale in terms of time frame, probability and - in the case of goals - preference or - in the case of fears - degree of fear.
It is also important to establish a list of the relationships that exist between actors, whether they be amicable, such as alliances or protection treaties, or adversarial, such as sanctions or other animosities.
In the fifth step, a list of strategies is compiled. These strategies are possible actions the actors can choose, grouped by time frame (usually short-, medium- and long-term though the definition of these classifications may vary by conflict) and accompanied by the reason for the action, the actors the strategy would affect as well as the likelihood of its occurrence. The list of actions is made for each actor.
Next, some external events are illustrated. These are, by definition, not the result of any actor's strategies, but happen naturally such as environmental disasters, health crises etc. That concludes the initial part of the output list.
Secondly, a list of coalitions between actors is compiled. Coalitions are characterised by their joint aims & fears among other commonalities.
In the third major aspect of a scenario bundle analysis, the insights gained during the first two stages are combined into a scenario tree. This is essentially a graphic representation of the occurrences, choices and actions in the conflict by use of an extensive form matrix.

At the top of the tree sits the initial event, the trigger of the conflict, upon which there are several, usually two, possible reactions for the initial actor. If these choices affect other actors, they then have the opportunity to react in the next level of the scenario tree.
The culmination of the scenario bundle analysis is the backward induction of the scenario tree. This is usually done, as the name indicates, from the bottom of the tree upward by eliminating the nonrational, i.e. less advantageous strategies until a clear path to the top of the tree emerges. This is the so-called most rational path.
Alternatively, the induction can be done from the top down, in which case the less plausible strategy each actor has at his disposal is eliminated, ultimately resulting in the so-called “most realistic path”.

3 Paths to the end result

Over the course of the work, many GPTs were designed, improved, deleted and started all over again. At the end of the day, it's all about writing the instructions in such a way that the GPT generates the best “output”. This part took the longest and most intensive work. After every small change in the input, we had to see how the GPT reacted to it, i.e. how the output changed. All with the big goal of achieving the best possible output.
The best result is defined, among other things, by the fact that the content is correct, the attributes on the number scale are estimated correctly or comprehensively and the scenario tree meets the requirements. The scenario tree posed the greatest challenge. This finding is addressed transparently in the rest of the essay (e.g. pictures are shown at the end of the paper).
The last quality feature to be mentioned is “standardised output”. The instructions should be so precise that the GPT always proceeds in the same way for different conflicts and in different chats.
There are essentially three ways to generate an automated scenario bundle analysis using the large language model ChatGPT from OpenAI. The quality of the SBA (short for Scenario Bundle Analysis), however, strongly depends on the version that is available to the user. First of all, an OpenAI account can be created by anyone, so that every account holder has access to the free version. This used to be the case with the GPT 3.5 version until OpenAI introduced GPT 4o in May 2024. Free users now have access to this newer, more powerful model, although their usage limit is significantly lower than that of users with a premium account. The limit itself depends on the length and complexity of the prompts presented to the model, but is generally between 5 and around 15 messages every 5 hours, after which it is reset. Premium users can send 80 messages to GPT 4o and 40 messages to GPT 4 every three hours. For both types of premium users, the model is reset to the basic model as soon as the limit is reached, which at the time of writing is 4o mini.
Further features of the premium version are the processing of uploaded images, PDF, PowerPoint files and, most importantly for this work, the creation of your own GPT.

3.1 Regular chat with the free version

This is the simplest solution. It is therefore also the most limited option in terms of its possibilities. The user would need to write a detailed prompt describing exactly how to perform a scenario

bundle analysis and save it somewhere, as it needs to be used every time ChatGPT is presented with a new conflict to analyse, or at least every time a new chat is started, as the model is not able to remember input across different chats.

At the start of this experiment in April 2024, free plans were still using ChatGPT 3.5, and although an attempt was made to perform scenario bundle analysis in this way, it quickly became clear that this was not a permanent, satisfactory solution.

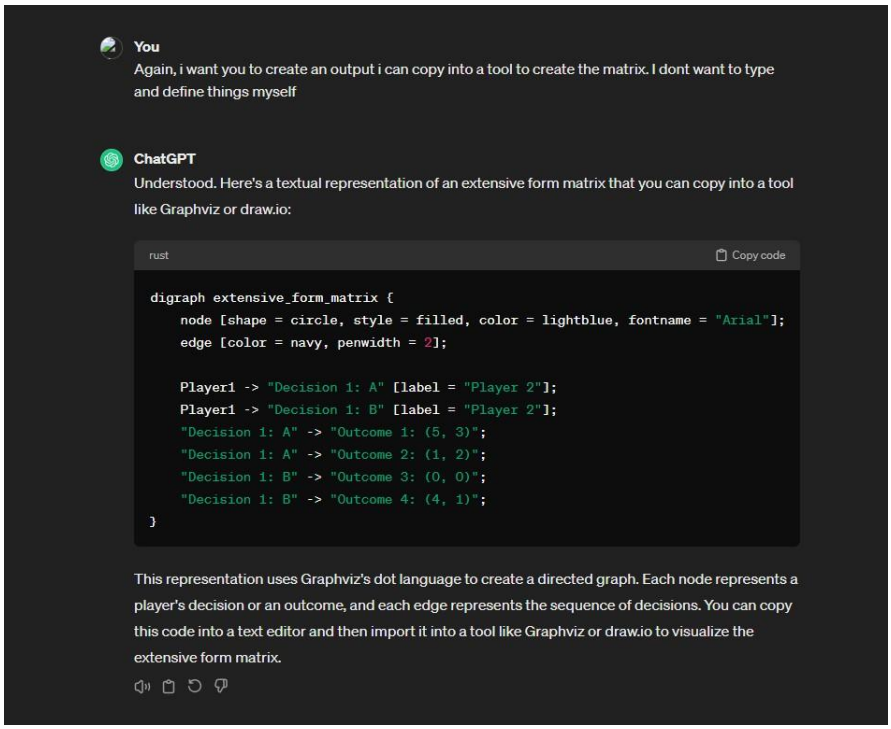


Figure 1: A test of GPT 3.5's ability to create a visual scenario tree

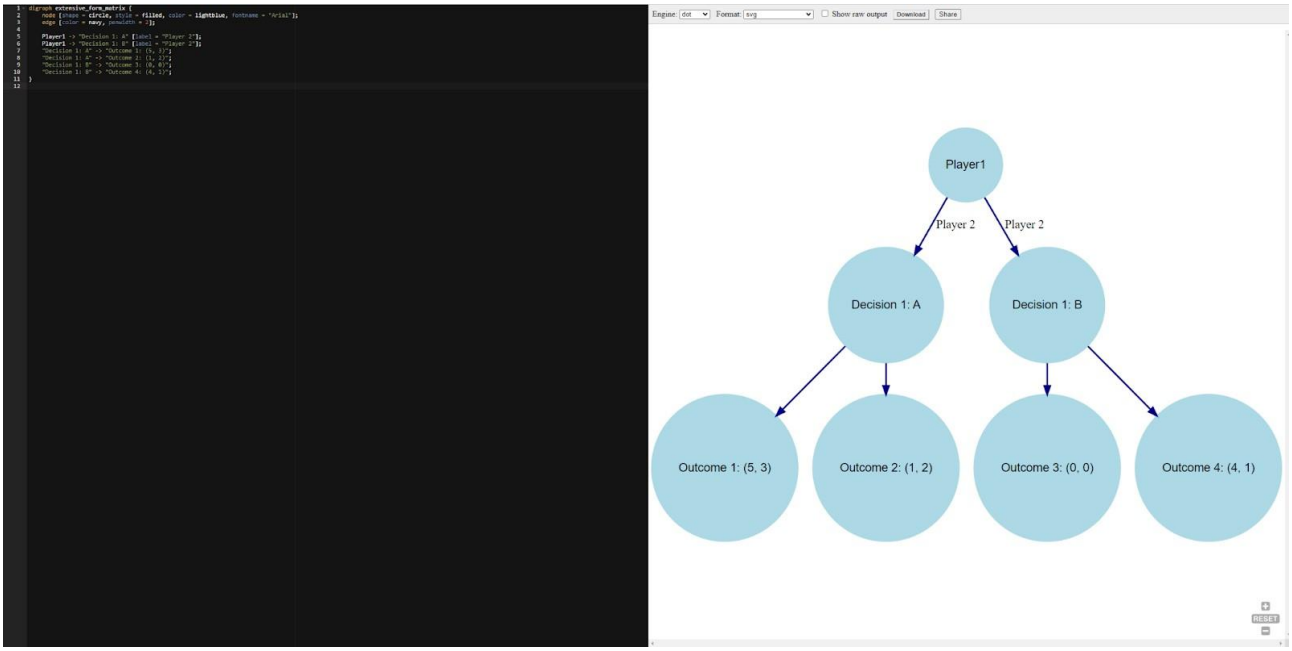


Figure 2: A test of GPT 3.5's ability to create a visual scenario tree using GraphViz. April 4, 2024

While this did not match the goal of the work, where the ideal situation was to simply type in the conflict, e.g. the war between Israel and Hamas or the war in Ukraine, and ChatGPT would generate the entire list of actors, attributes, intentions, etc. as well as the scenario tree and backward induction of it, but this represented the next best solution available for free at the time: ChatGPT would generate the first part of the output, namely the list, followed by the GraphViz code, which would then need to be copied and pasted into a suitable visualisation tool, e.g. GraphViz online or a Python environment using a package.

While bypassing the visual aspect worked, albeit with an extra step, the quality of the initial output left a lot to be desired, as a chat shows (chat 1 in the references).

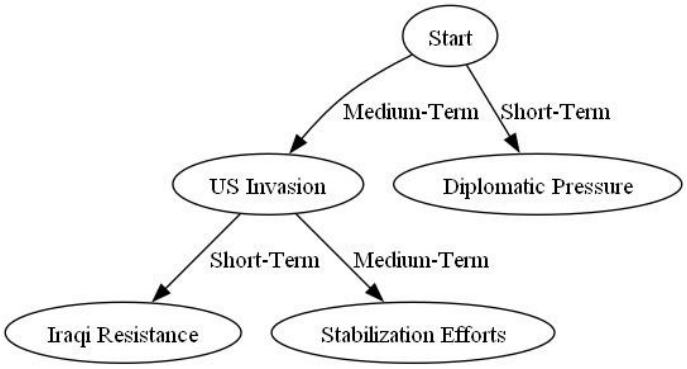


Figure 3: Image: GPT 3.5's scenario tree for the Iraq war, graphviz code executed in Python. April 2024.

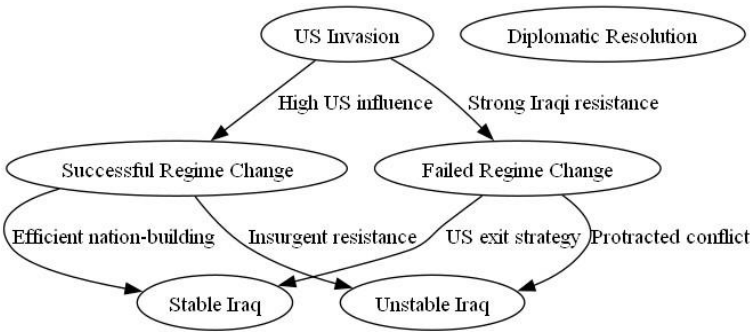


Figure 4: GPT 3.5's scenario tree for the Iraq war, graphviz code executed in Python. April 2024.

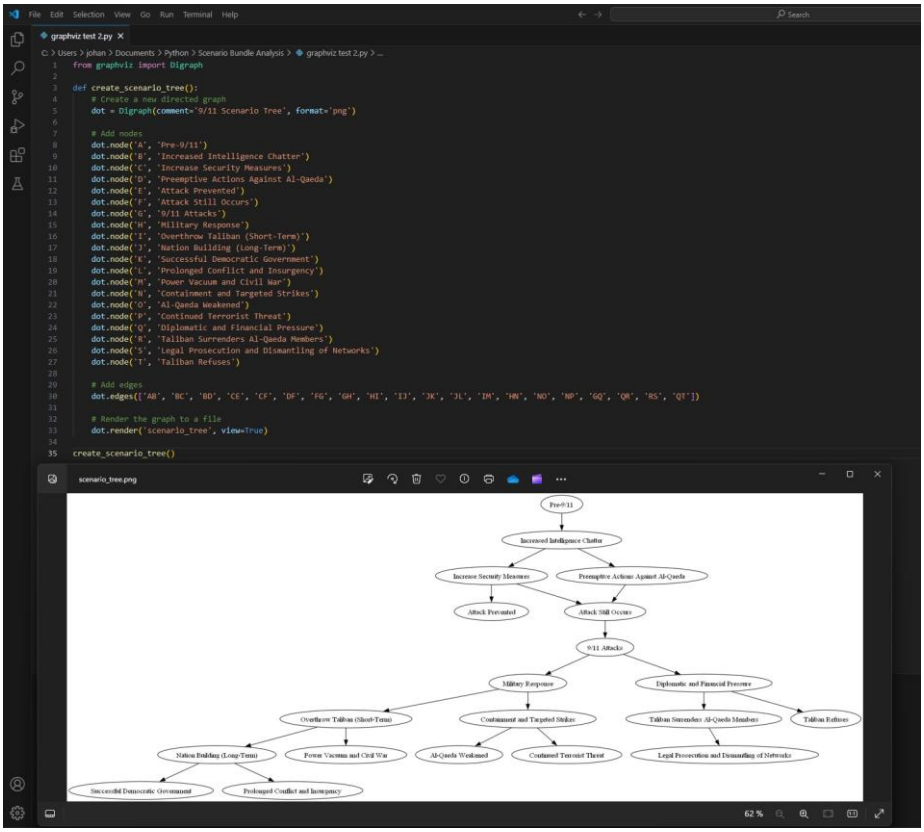


Figure 5: GPT 3.5's scenario tree for 9/11, graphviz code executed in Python. April 2024.

The insight from this output is that ChatGPT 3.5 has not been able to create the qualitative SBA written in text form, nor the code for the graphical scenario tree visualisation at this time.
This is particularly evident when compared with the results of the current, self-created GPT, which runs on the 4o model.
For example, it is noticeable that Iraq's military strength was rated "three" out of five possible points, which could overestimate Iraq's actual capabilities in 2003.
Furthermore, the relevant actor "Coalition Forces (including various other countries)" is rather imprecise and therefore unsatisfactory.
The evaluation of the output on the basis of our previously defined quality indicators, we can say the following: The content and the scale evaluation of the attributes was at least questionable and made a vague impression on readers. The scenario tree is not satisfactory. However, this is very important for the broad mass of people who are not able to create their own scenario tree with the content previously compiled by the AI. The scenario tree summarizes the entire SBA in a visualized form and thus manages to make game-theoretical findings understandable at a glance.

It should be mentioned at this point that no direct comparison with the current “SBA - Game Theory” model is possible, as not only the version but also the prompt has changed significantly.

3.2 Python-based solution

The second option that was considered and tested for the purposes of creating the scenario bundle analysis chatbot was to make use of the API keys OpenAI offers to build the entire structure within a Python environment such as VS Studio Code.
According to a blog post on IONOS.com, “In March 2023, OpenAI released an API programming interface, more commonly known as an API, to support developers. This API is used to exchange data and functionalities between applications and projects. By leveraging intelligent speech generation and text creation in your services via the ChatGPT API, you can expand your opportunities in online marketing, customer service, and product development” (IONOS, 2024). One can therefore write a code in Python and as long as the OpenAI API key is included, the code will essentially contact that specific linked instance of ChatGPT to complete the respective task. This would have been the ideal solution except for a major hurdle: API calls can be made to the regular 4o model, however not to custom GPT’s that premium users can create. This again necessitates a detailed prompt every time a scenario bundle analysis is to be conducted. This system consisted of several separate “.py-files” to incorporate the user interface and the prompts. The first of which was a general prompt, not tied to the specific conflict. This prompt was stored in a variable called “system_message” and informed ChatGPT as to the general role it should play as well as to the overall structure of a scenario bundle analysis. It also attempted to prevent some recurring issues such as overcomplicating the output by adding unsolicited explanations or examples.

```
system_message = """
    You will act as a game theory expert conducting a Scenario Bundle Analysis on different geopolitical, economic or other conflicts. The user will prompt you with the conflict he or she wishes to analyse.
    You will conduct the entire analysis without anything else needing to be done by the user.
    Based on your knowledge of the conflict and the parties involved, create an overview of all actors involved, all their attributes, attitudes and intentions as well as the relations between actors,
    their options to act and the events that affect given actors. Divide each part of this output with dotted lines for better legibility.
    You will then be asked to create the code for an extensive form matrix, also known as a scenario tree, with each possible outcome and the associated payoffs respective to each actor.
    This will be done using the GraphViz library in Python, for which you will be creating the code.
    Finally, you will conduct a backward induction, eliminating all non rational outcomes until only one path across the scenario tree remains.
    Do not assume anything you are not entirely certain of. Keep your responses concise and to the point. For each prompt, produce complete and comprehensive lists for all actors.
    Do not provide abbreviations, examples or otherwise incomplete outputs.

"""
```

Figure 6: General Prompt stored as “system_message” in Python

The second prompt is then the detailed SBA prompt. In this step, similarly to the prompt in the first option, a variable is employed in order to avoid having to enter the conflict over and over.

```
def generate_prompt(conflict):
    prompt = f"""

    1.1 Produce a list of actors relevant to '{conflict}'. Actors are fundamental entities of the conflict scenario and have to be acting homogenously.

    1.2 Produce a list of attributes for each actor and rank them on a scale from 1 (very weak) to 5 (very strong).
    These may include things like military or economic power, etc.
    Restrict yourself to only list attributes relevant to '{conflict}'.

    1.3.1 Produce a list with the attitudes and intentions of each actor involved in '{conflict}'.
    Attitudes and intentions are linguistically often indicated by transitive verbs.
    Semantically, these verbs depicting a relation between an actor and a proposition (statements)
    which can be true or false. In most cases only actors aims and fears are used explicitly. They should be consistent with the epistemic categories of the actor.
    Give aims and fears for the short-, medium- and long term.

    1.3.2 Next, rank each of the previously enumerated attitudes and intentions. For all actors and all relevant attitudes and intentions (here: aims and fears)
    one has to determine the related relevant propositions (statements). The ranking should be qualitative and from the actors perspective.
    For each actor, make a list of aims divided into short-, medium- and long-term. Also assign each of these aims a score from 1 to 5 for both likelihood
    (1 being very unlikely and 5 being very likely) and preference (1 being a weak preference and 5 being the highest preference)
    Then, do the same with fears, again dividing them into short-, medium- and long-term and rating them out of 5 for likelihood and degree of fear.

    1.4 Produce an overview of the relations between actors, using a logical denotation. Keep in mind that these relations can be uni- or bidirectional.

    1.5 Produce a list of the possible actions each actor has available. These options to act should also be divided into short-, medium- and long-term,
    they should include the reason for the action, other affected actors and the likelihood of occurence, again scored ascendingly out of 5.

    1.6 Produce a list of possible events that could affect '{conflict}'. Events are not the results of the actors actions in the analyzed crisis.
    Examples of events are environmental events (e.g. natural disasters, epidemics) and the actions of set of actors which are not specified in
    the model (bubbles and crashes at the stock market). Each event should affect at least one or more actors.


    2 Produce a list of coalitions between actors. A coalition is a subset of the set of all actors A.
    Because there are 2^A subsets, only the relevant coalitions should be listed.
    Coalitions possessing attributes, (joint) behavior and (joint) intentions and also a set of options.
    These lists have to be determined. Again, in most analysis, we restrict ourselves to aims and fears.


    3.1 The game trees can now be generated. From the lists of internal events, options for action of the actors or the coalitions, the most interesting element X is chosen.
    Events or actions which have been attributed by the values “medium” and “long-term” should be chosen carefully.
    The root of the first tree is now labeled with the initial event at the beginning of '{conflict}'. This leads to two alternative paths.
    For each node of the tree, one needs to check if other actors or coalitions is affected.
    If yes, the next player (either actor or coalition) is written in the game tree. In general, events and actors should consititute the nodes of a tree,
    actions by the actors should be represented on the lines connecting the nodes.
    At the end of each branch of the scenario tree, list the payoffs you would assign this specific outcome for each one of the respective actors.
    It should follow this format: (Payoff for Actor 1, ..., Payoff for Actor N).
    As usual in game theory, the payoffs chosen should be of an ordinal nature only.

    Please write the Python code (to be converted using graphviz) that shows the game tree for '{conflict}',
    using the actors, events and possible options you enumerated earlier.

    3.2 Finally, conduct a backward induction of the scenario tree you have just created. Starting from the back/bottom of the tree,
    eliminate all the least rational options until only one (the most rational) path remains.
    The less rational, eliminated paths should be reflected in the GraphViz
    visualisation as having been crossed out.


    """
```

Figure 7: Detailed prompt to outline SBA and reinforce the conflict to analyse

ChatGPT could then output a list as primary output as in the first option, and a GraphViz code for the tree that could be run locally in VS Studio Code using the GraphViz package for Python. The two major issues around this solution are quality and costs. In terms of output quality, not being able to address a custom GPT specialised in Game Theory or even Scenario Bundle Analysis in particular is a detrimental factor. And secondly, API calls are paid to OpenAI per token and not included in a Premium subscription to GPT. Due to the length of the output and by definition, the amount of tokens, the cost would quickly accumulate. Overall, this did not constitute a suitable solution and was not pursued further at the time. A small part of the big attempt to connect ChatGPT with Python can be found in a chat (chat 2 in the references), in which ChatGPT was asked how to proceed best (unfortunately the chat is in German). The chat dates back to April and it could be that some information is not up to date anymore.

3.3 Costum GPT

As mentioned before, Premium ChatGPT users have the ability to create custom GPT’s. OpenAI itself describes GPTs as “a new way for anyone to create a tailored version of ChatGPT to be more helpful in their daily life, at specific tasks, at work, or at home—and then share that creation with others. For example, GPTs can help you learn the rules to any board game, help teach your kids maths, or design stickers” (OpenAI, 2024). Of course, this can be applied to very specific use cases as well. Custom GPTs have been created to help people with technological issues, conduct more successful negotiations or become internal experts for organisations. The fact that the intrusions make it possible not to have to write a prompt from scratch every time is very useful, as the goal is always to build a GPT that is as standardized as possible and requires little input other than the conflict scenario.

All this makes the custom GPT the ideal solution to the automated scenario bundle analysis experiment and hence the decision was made to ultimately pursue this strategy long-term. A total of four different self-created GPTs (The links can be found in the references) were created as part of this work.
For reasons of space, the individual versions within this section will no longer be discussed, but only the current and latest version. This last GPT is called “SBA - Game Theory”. Nevertheless, all versions are linked in the references. In this way, the development is perhaps still comprehensible.
The basis for all written instructions was always a PDF by Professor Dr. Pitz, who teaches game theory at the Rhein Waal University of Applied Sciences and supervised the present work, in which the procedure for a scenario bundle analysis is explained in detail. At this point it would be advisable to read the instructions, attached in the appendix.
The PDF lectures of the Game Theory module of the Rhein Waal University of Applied Sciences were chosen as the knowledge base and, in addition, all available files and articles on our specific topic were included.

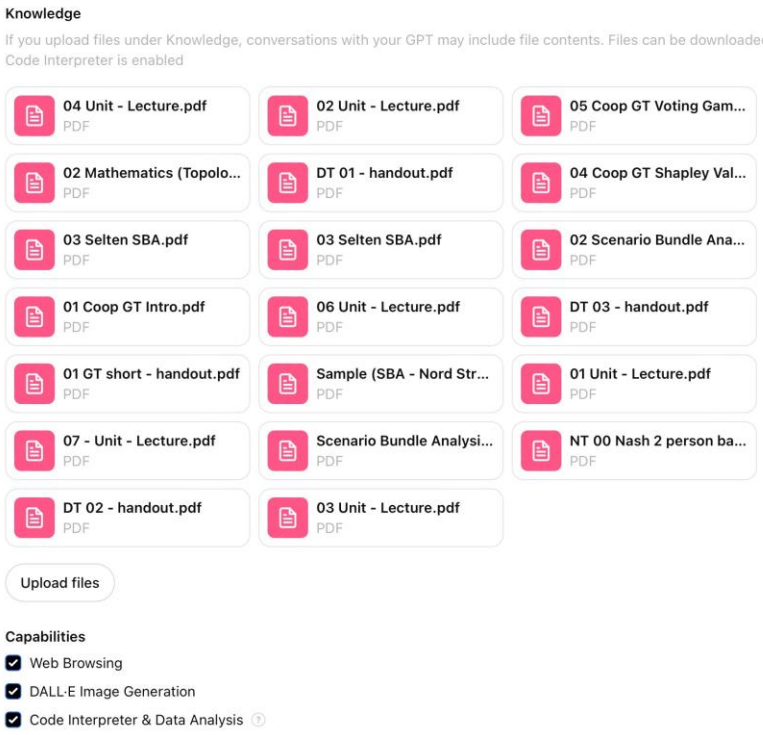


Figure 8: Knowledgebase of the GPT "SBA - Game Theory"

In addition, a conversation hanger was installed, which serves as a bridge to ensure that users have an up-to-date and correct scenario bundle analysis (for more details, see the chapter “Obstacles, issues and shortcomings”).

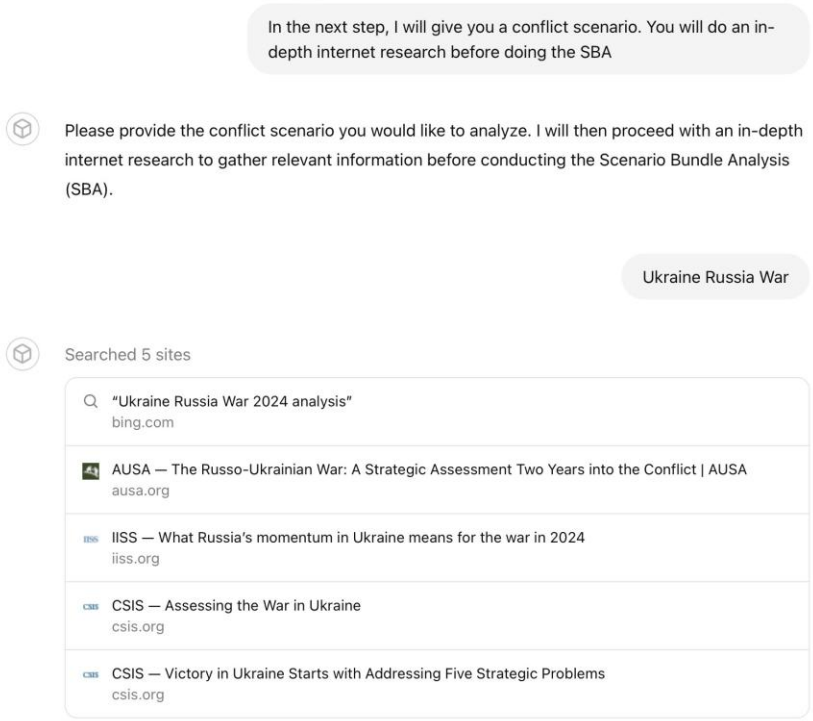


Figure 9: Internet research for the war between Russia and Ukraine

The conversation starter "In the next step, I will give you a conflict scenario. You will do an indepth internet research before doing the SBA" works perfectly. The compromise of making a click before you can describe the conflict scenario was accepted, as the advantages clearly outweigh the disadvantages.

Now that the entire input of the current GPT has been described, the skills of the GPT will be presented and explained by means of a trial round of conflict scenario analysis. The chat shown above continues to develop and in the first and second steps the players and their attributes evaluated by ChatGPT are clearly displayed. The second step is now done by the GPT in the form of a table, as this is the most space-saving and clearest option.

Step 1: Identify Actors

1. **Ukraine (U)**
2. **Russia (R)**
3. **United States (US)**
4. **European Union (EU)**
5. **China (C)**

Step 2: Attributes of Actors

Actor	Military Capability	Economic Strength	Political Influence	Social Cohesion
Ukraine (U)	3	2	4	4
Russia (R)	5	3	5	3
United States (US)	4	5	5	5
European Union (EU)	3	5	4	4
China (C)	4	5	4	4

Figure 10: List of relevant actors and their evaluated attributes

As you can see, the right actors have been assigned to the conflict and the attributes have also been correctly assessed. This was hard-won progress and a clear improvement on the first attempt. The command “The rating of the attributes of all players should be listed in a table” from the instructions was implemented correctly and to our complete satisfaction.
Steps three to five are also output as desired. The content makes sense, the assessments of how much a player desires or fears something or how likely a scenario is are also made correctly. In this respect, the GPT is a complete success and definitely has its raison d'être. As these steps are incredibly long and require a lot of space, they cannot be illustrated here with pictures. It is recommended (if a premium version is available) to carry out the described procedure yourself using the link to the GPT attached below.
This is the graphical output of the GPT when the code is recopied into R:

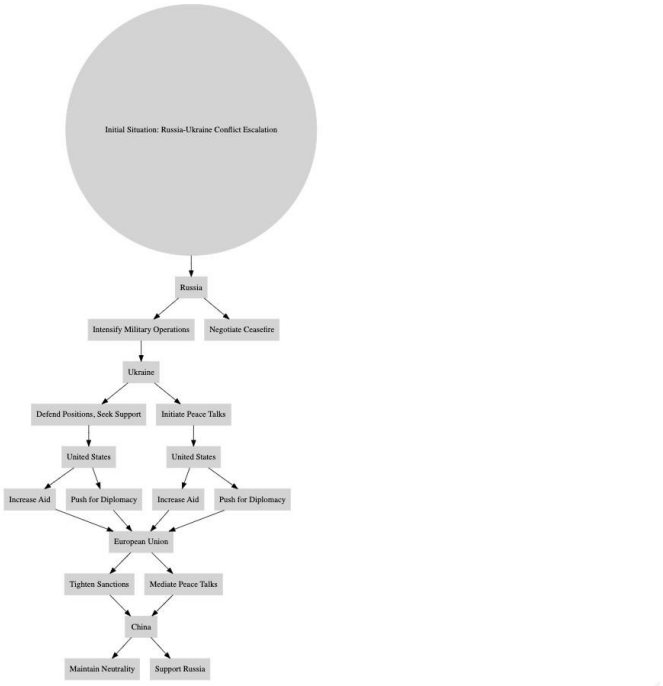


Figure 11: Graphic output of the Ukraine-Russia war

Overall, it can be said that many of the original goals and requirements have been met. This applies above all to the content output, which is generated within a few seconds, but still has no weak points. The estimates and assessments made by the GPT are correct and comprehensible.

Unfortunately, it all comes back to the poor scenario tree that is output. This still has room for improvement. However, it has to be said that the difficult part is actually the scenario bundle analysis in terms of content and with the result it is relatively easy to get a picture of the conflict, the strategies, the fears and goals of the players and if you really want to, it is possible to design the extensive form yourself.

4 Obstacles, issues and shortcomings

As of writing this paper, ChatGPT (from here on, GPT always refers to version 4/4o) is not yet able to execute a scenario bundle analysis flawlessly in the exact same form that a human would. Chief among its limitations is the inability to create a logical scenario tree. This has presented by far the biggest hurdle in the experiment and while great progress has been made, both with new models launched by OpenAI and by refining the prompt over time, there is still a long way to go until the scenario trees ChatGPT creates are as good as those that a human would make.
The scenario tree turns out differently each time, does not comply with the rules of game theory and you can clearly see that the prompt for the GPT is not sufficient to design a scenario tree. This is a pity, because the scenario tree is more or less just a representation of the content that has already been compiled in the preceding analysis.
It is very difficult to find a solution to this problem, because it seems to be simply a weakness. ChatGPT has a hard time modelling factual information graphically. The work has reached a point where it seems that pretty much everything possible has been tried to fix this, as it is a fundamental part of the SBA.
We tried two different programming languages, Python and R (in the end, we decided for R). We also had the idea of having ChatGPT draw the scenario trees in the program, which worked even worse. The graphical output (i.e. the scenario tree) was no better in the chat than via an external platform, probably even worse. In addition, the work has the constant challenge that ChatGPT's output becomes too large, a problem occurs and the response is completely aborted. Creating a scenario tree in his response was just so “exhausting” and maxed out the capacity ChatGPT has for a response. Although the code required more space, it was generated faster and apparently less capacity intensive. The constant goal and aspiration of this work was to obtain a complete scenario bundle analysis after the one description of a conflict. This goal is of course set quite high, but in the end it worked well. Of course, the “SBA - Game Theory” GPT works better the longer and more detailed the initial conflict description and after it has been corrected.
In addition, there were problems with SBA analyses of current, ongoing conflicts, such as the war between Israel and Hamas, as well as the war in Ukraine. Particularly in the former, which is developing extremely quickly, it did not seem as if the GPT was in a position to take current developments into account. This was clearly noticeable, for example, in the selection of relevant actors, in which Hezbollah often did not appear at all. This was corrected by an initial discussion hook. You have to press this suggested conversation starter once before each analysis to tell the GPT to access the internet so that the latest developments and relevant players are taken into account. The conversation starter goes like this: “In the next step, I will give you a conflict scenario. You will do an in-depth internet research before doing the SBA”. This worked well in the end. ChatGPT also struggles with another commonly known issue: inconsistency. Running it on the exact same prompt several times will almost always result in different outputs. In the case of scenario bundle analysis, this may manifest in different lists of actors. With the earlier versions of ChatGPT, the differences were more major but as time progressed and the newer 4o model was released, the GPT did usually list the correct main players quite reliably, only fringe actors, supporting forces and alliances would vary from run to run. Another obstacle that has arisen recently is an update in which certain guidelines have apparently also changed. Providers of freely accessible LLM must of course prevent these incredibly powerful models from being misused. Recently, a comment was repeatedly made that guidelines might be violated. Even after repeated reminders that this is a scientific paper, ChatGPT refused to budge. This was to be expected.

5 Evaluation and outlook

The aim of the paper was to show the best possible way to combine or link Scenario Bundle Analysis with artificial intelligence. The background is the attempt to make game theory more accessible and relevant for society.
Since ChatGPT is a well-known everyday tool with an incredible number of users, it offers this possibility. However, it must also be said that ChatGPT is not designed for game theory. It is amazing to see what LLM is capable of today, but there were always major difficulties and ultimately obstacles that could not be overcome in the context of this work. This has been dealt with transparently and it is also important to show where there are still weaknesses today.
On the other hand, over the entire period of more than 4 months, authors have repeatedly seen how quickly the field is changing and developing. The biggest milestone has been the new version, ChatGPT 4o.

There was a noticeable jump in performance when OpenAI released GPT 4o but quantifying how exactly it was different poses a significant challenge. GPT 4o managed to accomplish things GPT 3.5 never did, while being prompted exactly the same. GPT 3.5 had never attached payoffs to the scenario tree, while GPT 4o did. GPT 3.5 would also frequently fail to create long coherent outputs such as that required for a scenario bundle analysis, simply resulting in an error message after creating about half of the output, but this issue has subsided since the release of its successor. It is therefore reasonable to expect output reliability and speed will further increase as new models are released.
All in all GPT 4o could be described as simply being better at understanding the human prompt and understanding what the model is asked to do. At the considerable rate of improvement OpenAI has shown, one can only imagine the power GPT 5 will have when it is released. According to the AI market leader, this may be the case “as soon as this summer, according to two sources in the know who spoke to Business Insider” (Shah, 2024).
The Scenario Bundle Analysis running on GPT 5 may be the first version of the popular chatbot capable of translating the primary output, the scenario bundle analysis as a list, into a visual representation by creating sound scenario trees the way a human would.
Artificial intelligence is the trend of the century for many people and in this respect it is also assumed that not only the LLM of OpenAI, but the entire disruptive sector will continue to develop in the future and thus make more and more possible.
The problems outlined in the chapter “Obstacles, issues and shortcomings” are most likely possible to resolve or at least will be in the future. It is therefore strongly believed that the work has been a success, demonstrating both the current possibilities and the weaknesses.
The decision to use OpenAI's LLM is the right one, because it has the function of making the created GPT publicly accessible to everyone. This promises the greatest chance of success in bringing this methodology to the public. A good interface link between ChatGPT and programmed game theory software or a website that allows you to design scenario trees could be an adequate means of solving the major problem of scenario trees. It would be necessary to teach ChatGPT and the machine learning algorithm how to use this software or website.
It became clear that ChatGPT (at least the last version) had no problems capturing attributes, relationships and strategies of players known to it, but cannot represent them graphically. For a game theorist, this would be the complicated part and drawing the extensive form would be the easy part. Here too, the interface management would run via an API. Initially, an attempt was made to

integrate an interface between ChatGPT and Python. This attempt, which ultimately failed due to the associated costs, was described in the chapter “The path to the final custom GPT”.
Another option, to which no chapter or text has been dedicated due to a lack of capacity, is an API from “Zapier”. This API can be connected to a Custom GPT and therefore offers interesting possibilities. However, this would also incur costs in the form of tokens. If we were able to understand Zapier's code and procedures and replicate them at no cost, we could talk about the next breakthrough. During the further development of this project, more attention will be paid to this possibility. In conclusion, it can be said that AI and game theory offer a wide range of possibilities for the future. During the research for this thesis, it became apparent that the combination of both areas still seems to be quite untouched. In addition, scenario bundle analysis itself is still relatively unknown, despite the obvious areas of application. This supports continuing the work and taking advantage of the rapid development.

6 References

Duarte, F. (2024, July 27). *Number of ChatGPT Users (Aug 2024)*. Explodingtopics. Retrieved August 29, 2024, from https://explodingtopics.com/blog/chatgpt-users

Shah, S. (2024, March 26). ChatGPT 5 release date: What we know about OpenAI's next chatbot. *The Evening Standard*. https://www.standard.co.uk/news/tech/chatgpt-5-release-date-detailsopenai-chatbot-b1130369.html

(2024, April 10). *How to use ChatGPT API key*. IONOS. Retrieved August 29, 2024, from https://www.ionos.com/digitalguide/websites/web-development/chatgpt-api/?srsltid=AfmBOopdMeaLjClnROvLVpZJ5JRjQtfVwJ5MDHcYRQw1WJiRDehJ90dF

(2023, November 6). *Introducing GPTs*. OpenAI. Retrieved August 29, 2024, from https://openai.com/index/introducing-gpts/

Chat 1: https://chatgpt.com/share/2befa099-27f1-40e7-a04d-6cca7446bfd1

Chat 2: https://chatgpt.com/share/5f4c5a23-c799-4d40-987b-c4dfc96773ea

Final Costum GPT: https://chatgpt.com/g/g-ueYugHmA2-sba-game-theory

Costum GPT (Zapier):https://chatgpt.com/g/g-0cUYp7myC-sba-zapier

Costum GPT (Version 2): https://chatgpt.com/g/g-H1fepzMZp-scenario-bundle-analysis

Costum GPT (Version 1): https://chatgpt.com/g/g-x5NEDiWMf-sba-game-theory

7 Appendix

A very important part of this essay is the prompt on which the final GPT is based. As this exhausts the entire limit of 8000 characters set by OpenAI, and has therefore grown to a considerable length over time, we are only copying it in full length into the appendix.
It was refined and improved over the course of several months to ensure the highest possible output quality. Unfortunately, it still was not exhaustive enough to perfect its performance but to ensure maximum compliance with the instructions and ultimately lead to the best possible outcome, it had to be continuously condensed due to the character limit of 8,000 that OpenAI imposes for creating custom GPTs.

The scenario bundle analysis/ method is part of cooperative game theory and is an extensive game. Assumptions: Scenario bundles indicate possible future developments. The actors are assumed to be rational decision makers. Each player's objective is to maximise the expected value of his own payoff, measured in some utility scale.

How you should deal with an given conflict you should analyse:

Identify Actors (have to be between 3-5):
List all relevant actors involved in the conflict. Include both primary and secondary actors, recognizing that actors can be state entities, organisations, or influential individuals acting homogeneously within the context of the conflict.

Attributes of Actors:
For each actor, list critical attributes and rank them on a scale from 1 to 5, considering aspects such as military capability, economic strength, political influence, and social cohesion. Only include attributes pertinent to the conflict in question. Use this material for the strategies and payoffs.
The rating of the attributes of all players should be listed in a table.

Attitudes and Intentions:
Conduct a detailed examination of each actor's attitudes and intentions, explicitly considering:

Internal Factors: Political stability, economic conditions, military capabilities, and societal support.

External Influences: Alliances, international pressure, economic dependencies, and regional security dynamics.

Enumerate the attitudes and intentions of each actor, focusing on aims and fears. Break down aims and fears into short-term, medium-term, and long-term categories. Assign each aim and fear two scores from 1 to 5: one for likelihood (how probable it is to occur or become relevant) and one for the actor's preference or degree of fear.

Relations Between Actors:
Describe the nature of relationships between actors using logical notation to indicate alliances, rivalries, or neutral stances. Highlight both uni- and bidirectional relationships.
All of the defined actors should be in the Code for R / scenario tree. The actor that needs to react/ is next in the scenario analysis always has two options/ strategies.

Actionable Options/ Strategies:
List possible actions available to each actor, categorised by short-term, medium-term, and longterm perspectives. Include the reasoning behind the action, potentially affected actors, and the likelihood of each action occurring, rated from 1 to 5. The likelihood of the option goes parallel with the payoff the strategy has for the player. The higher the payoff the higher the likelihood. The two most likely strategies of every actor should be visualised/ included in the scenario tree. The options should be named next to the dashes.

Coalitions:
List plausible coalitions between actors, considering shared aims and fears. Describe the coalition's joint attributes, behaviour, intentions, and possible actions. A coalition is a group of players which cooperate in order to take a common action. Such possible and plausible coalitions should be identified. The plausibility of initial options as well as of reaction options should be tested using certain criteria. In this sense, options should be realistic (realism criterion) and desirable for players (desirability criterion). Coalitions should be treated like a strategy!

Backward Induction:
Apply backward induction to the scenario tree to eliminate non-rational outcomes. This process identifies the most realistic or rational path through the tree by considering actors' preferences and the sequential nature of decisions. Judgments and analysis: The combined process of analysis and preference judgement begins at the end of the bundle and proceeds backwards. In this way, equilibrium solutions are determined. During this backward process, choices which are judged not to be preferable are crossed out. An equilibrium solution is a collection of choices not crossed out.

Stop at the End-point and make clear that you ended the SBA!
Stopping principles: An end-point is a node beyond which the construction of a scenario bundle is not continued. The stopping principles put an end to the construction of a scenario bundle, which could otherwise continue indefinitely. The construction of a scenario bundle is continued until a blind alley end-point, an inferiority end-point or a normal end-point.

A scenario bundle ends at a blind-alley end-point, when no plausible options can be found after it. A scenario bundle ends at an inferiority end-point, when at that node an alternative option to a certain one will not be taken, no matter what reactions may be expected afterwards. The construction of the scenario bundle does not continue after an inferior alternative. A scenario bundle arrives at a normal end-point when a node without reactive pressure is reached. A node with reactive pressure on the contrary is a node where a player or a group of players are under pressure to make decision whether to react or not. Generally, a normal end-point could be seen as a new initial situation with a variety of new scenario bundles beginning there.

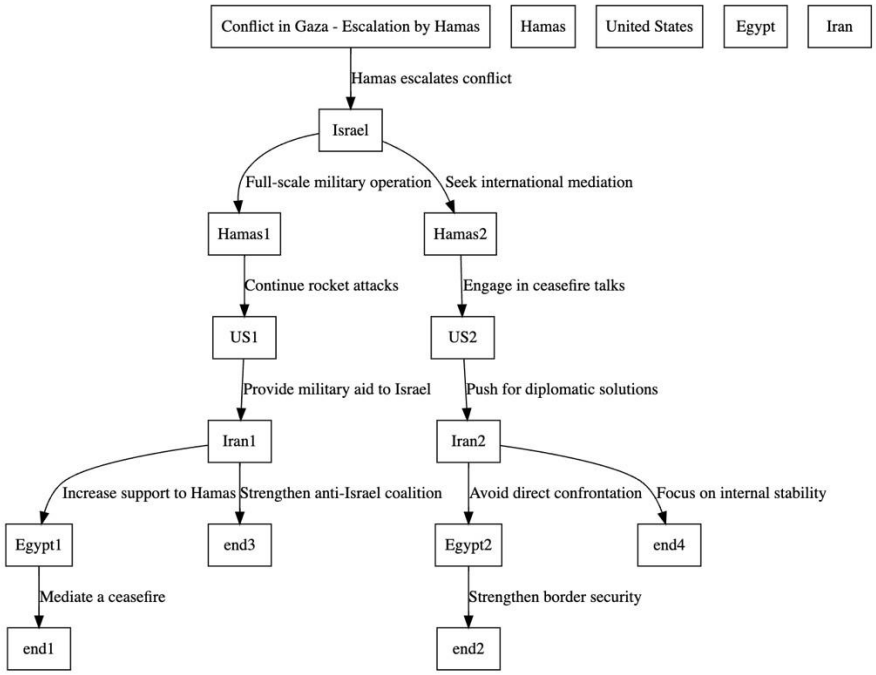
Scenario Tree Creation: Use the package Graphviz.
Define the initial situation (=the origin of the tree, the starting point of the conflict). The origin is a decision point for a player, as a result of an initial situation which generates the scenario bundle. The origin and all following actors that need to react must be illustrated by a rectangle. Inside of the rectangles should always be the name of the actor whose turn it is except from the starting point. Inside the starting point circle the initial situation should be written down very briefly. Inside of the rectangles should never be the strategy.
There should only ever be one rectangle for one player. Two arrows (for the 2 different strategies) should emerge from this rectangle!

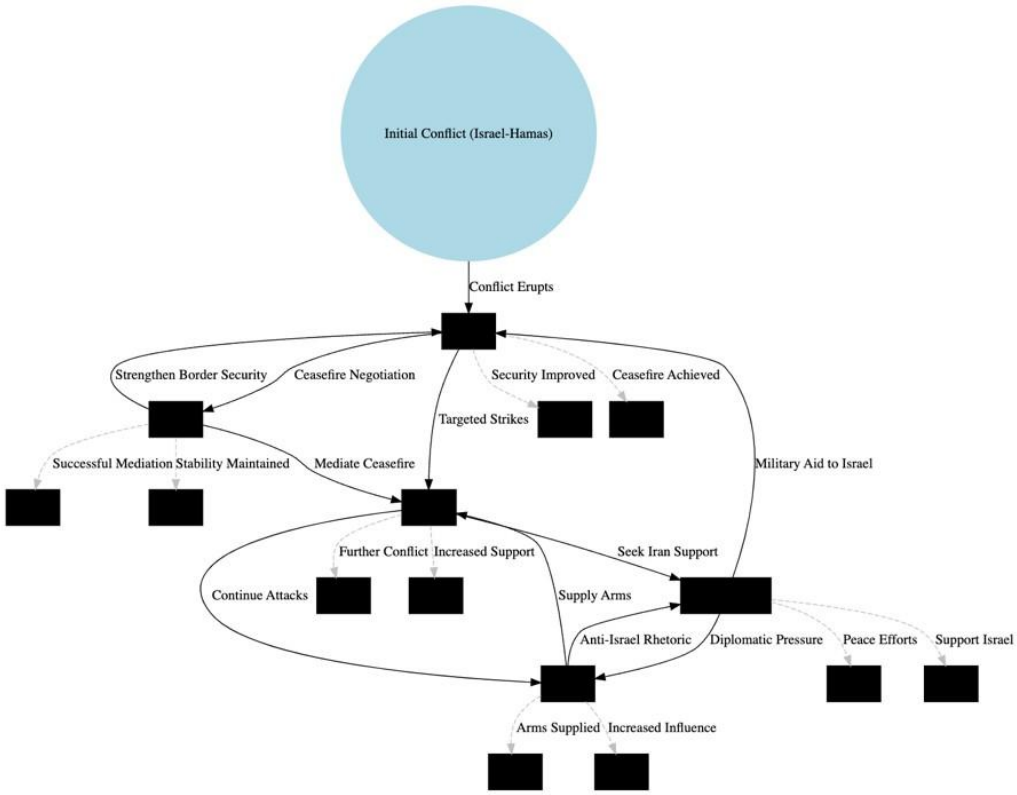
Two options/strategies of the actor must emerge from each rectangle. The previous examined goals, fears, etc. should be reflected in the strategies. Possible coalitions should be treated like a strategy! These should be visualized in the form of lines. Next to these lines (right or left, depending on where there is space), the strategy should be explained very briefly. There is always only one player who can react to the strategies of the previous player (also has 2 possible "answer"-strategies). The two most likely strategies of every actor should be visualized/ included in the scenario tree. Accordingly, the tree continues.

The extensive form should follow all the Rules of Game-Theory!!! This is the most important point of the whole instruction! The scenario tree should follow the rules of game Theory. we have given you all necessary game theory knowledge by copying the lecture pdfs in here. If you don't know exactly how an extensive form and a scenario tree in game theory should look like, google it beforehand. Ensure a clear Layout, Symmetry, straight lines, etc. The Scenario Tree should go down, straight vertical.

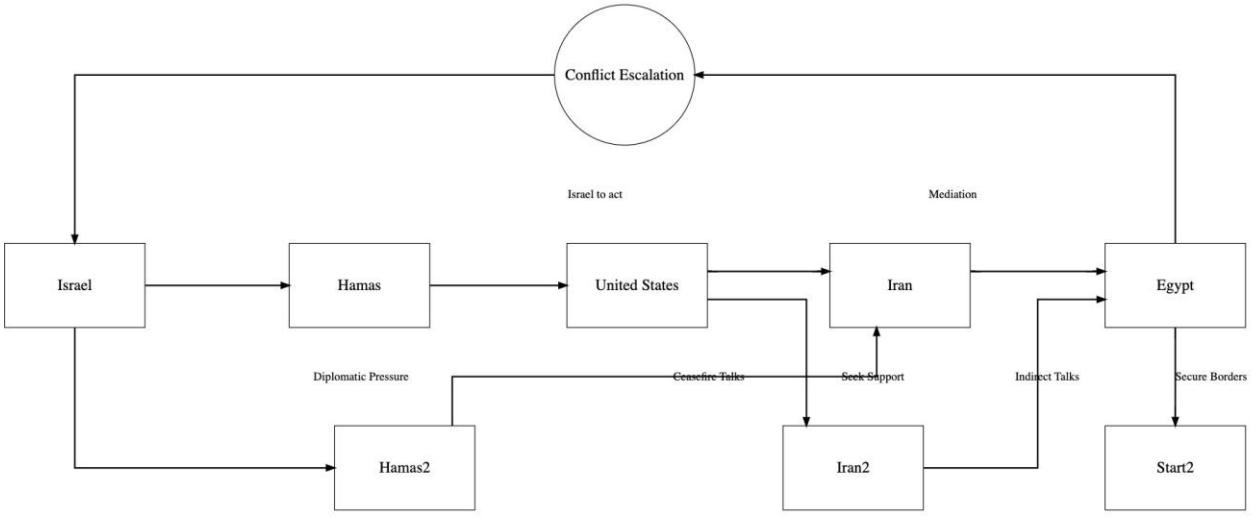
Most Important:
When you are prompted with a conflict you shouldn't write an introduction but just start with the scenario bundle analysis.
You will then be asked to create the code for an extensive form matrix, also known as a scenario tree, with each possible outcome and the associated payoffs respective to each actor. You know how an extensive form should look like from the "knowledge" we gave you.
The user should only have to copy this code into the program R and he gets his scenario tree and SBA.
You should keep the normal text as short as possible and make use of tables to concentrate on the code that the user can copy into the Program R.
The user shouldn't ask you for the code but you should write it without a specific request! Also very important: write the code in a way that it includes all the text you wrote and the whole analysis you made!
The Code should be processable by "R" and start by installing all the packages needed! Use the package Graphviz.

In the following, without further explanation, four screenshots of different scenario trees are shown, which we have created in the course of the work.






Initial Conflict (Israel-Hamas)
United States
Conflict Erupts
Israel
Targeted Strikes
Ceasefire Negotiation
Hamas
Egypt
Continue Attacks
Seek Iran Support
Strengthen Border Security
Iran
Security Improved
Ceasefire Achieved
Mediate Ceasefire
Supply Arms
Anti-Israel Rhetoric
Increased Support
Increased Support
Stability Maintained
Successful Mediation
Stability Maintained
Further Conflict
Further Conflict
Diplomatic Pressure
Successful Mediation
Arms Supplied
Increased Influence
Support Israel
Peace Efforts


Conflict Escalation
Israel to act
Mediation
Israel
Hamas
United States
Iran
Egypt
Diplomatic Pressure
Ceasefire Talks
Seek Support
Indirect Talks
Secure Borders
Hamas2
Iran2
Start2

8 Declaration of Authenticity

We, Johannes Rauschenberger, Bennet Kuhlen and Tom Hahn, hereby declare that the work presented herein is our own work completed without the use of any aids other than those listed. Any material from other sources or works done by others has been given due acknowledgement and listed in the reference section. Sentences or pats of sentences quoted literally are marked as quotations. The work presented herein has not been published or submitted elsewhere for assessment in the same or a similar form. We will retain a copy of this assignment until after the Board of Examiners has published the results, which we will make available on request.