**Group 11: Treasuring Land**

**Project Summary**

“Treasure Land” is a brave adventure game. The player has to follow the clues to find the treasure. However, in the process of searching for the treasure, there will be many difficulties, such as marsh, weather, or getting hurt for unknown reasons, which will lead to death.On the other hand, in the process of searching for the treasure, there will also be things happening in the favor of the player. For example, you will find an artifact, which is a weapon that will help you to get through the difficulties, and furthermore, you can solve puzzles to get the treasure that will make you perfect the game called “Treasure Land”.

**Propositions**

Bridge(x): A bridge exists at the location x. If we want to find the clue and there is a river, we can pass the river

River(x): A river exists at location x. A river can block the way to find the clue.

Puzzle(x): A puzzle exists at location x. We can find clues to find the puzzle, if we solve the puzzle we can find the treasure.

Hazard(x): A hazard exists at location x. If the player gets a hazard, the player can not find the treasure.

Artifact(x): An artifact exists at location x. An artifact can help the player avoid the hazard.

Clue(x,y): A clue exists at location x pointing to location y. As long as the player finds the clue, the player can find the puzzle through the clue point x given at point y.

Treasure(x): Treasure at location x. The winning state.

Rainy\_Wether(x) = True is the weather is rainy when the game is continuing.

Marsh(x) = True if the weather is rainy.

**Constraints:**

When there is a river and the player as location x does not have a bridge to pass the river, the player won’t have access to the clue in the location x.

River ∧ ¬Bridge → ¬Clue

If there is a river and we have a bridge, the player can have access to the clue at point x.

Bridge ∧ River → Clue

When the player has found the clue at location x, the player can find the puzzle at location y that location x has provided.

Clue → Puzzle

If we have access to the puzzle from the clue and we get no hazard then we can have the treasure at location x.

Puzzle ∧ ¬Hazard → Treasure

When a hazard exists and the player has no artifact, the player can not have access to the hazard.

Hazard ∧ ¬Artifact → ¬Treasure

If we have artifacts then we will not get into trouble.

Artifact→¬Hazard

When there is rainy weather, the location will be a mess.

Rainy\_Weather→Marsh

If the mersh appears, and there is no artifact, it will cause the player to not find the clue location.

Marsh∧¬Artifact→¬Clue

When the player does not have the bridge to pass the mersh, the player cannot have access to the clue location x.

Marsh∧¬Bridge→¬Clue

**First-Order Extension**

**Propositions for the First-Order order**

Player1(i): First player who joined the game.

Player2(i): Second Player who joined the game

Weapon: Players can use weapons to eliminate enemies.

Barbarian: These guys will attack you when they meet you

Barbarian\_Chief: can be found by barbarians. Players can learn the location of Zephyr by talking to him.

Zephyr: Appears randomly anywhere on the map and does not move. When the player meets him, you must link up with another player to challenge him. Successful challenges earn you a respawn.

Respawn: come back to life from where you died.

**Constraints:**

Barbarian at location x; Barbarian\_Chief at location x; Player1 at location i; Player2 at location j; Zephyr at location y.

The Barbarians cannot be defeated if no player has the weapon.

∀x(Barbarian(x) ∧¬Weapon → ¬Player(i) ∨ ¬Player2(j))

If either player1 or player2, location i, talks to the barbarian chief, location x, , they can know the location of Zephyr, location y.

∀x(Barbarian\_Chief(x) ∧ (Player1(i) ∨ Player2(j)) → Zephyr(y))

If either Player One or Player Two were missed, we can not get the respawn chance.

∀y(Zephyr(y) ∧ (¬Player(i) ∨ ¬Player2(j)) → ¬Respawn)

The Barbarian can be beaten if either player brings the weapon.

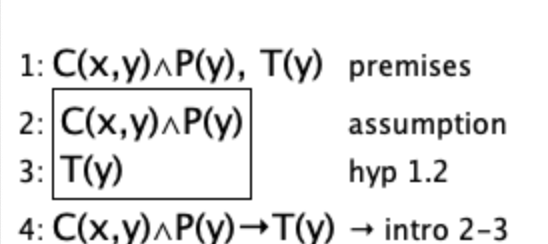
∀x(Barbarian(x) ∧ Weapon ∧ (Player1(i) ∨ Player2(j) ) → ¬Barbarian(x0)

Both players need to work together to defeat Zephyr so that they can earn a chance to respawn

∀x(Zephyr(x) ∧ (Player1(i) ∧ Player2(j)) → Respwan)

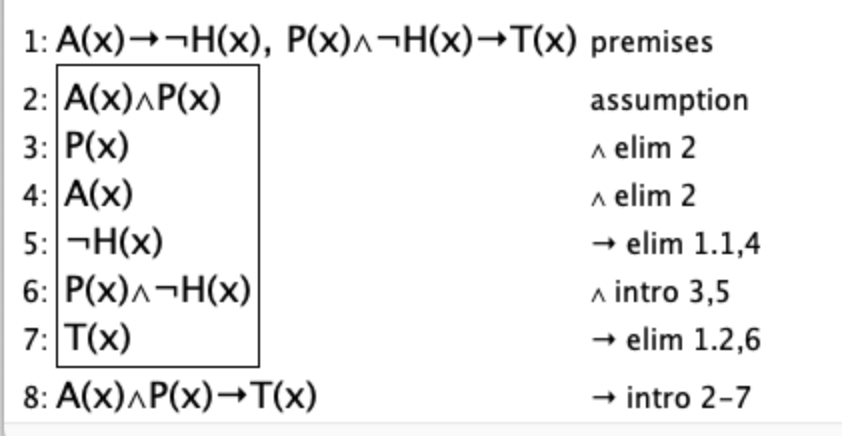
**Model Exploration**

**Jape Proof Ideas**

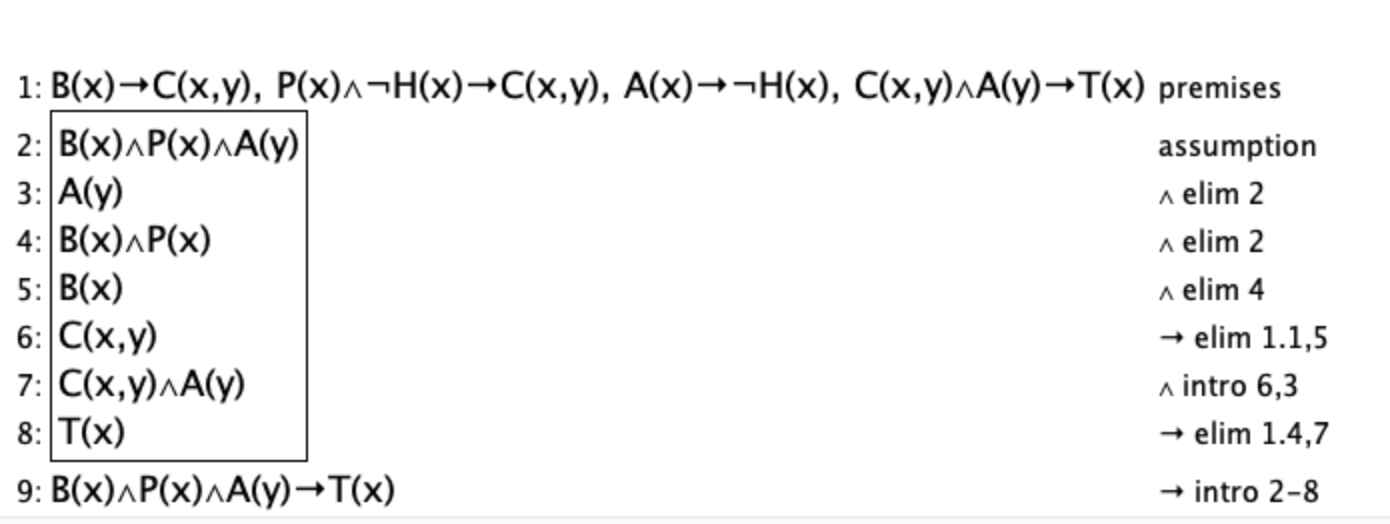


As long as we have found the clue, and then we can find the puzzle, and if the puzzle is true(Solve the puzzle) and then we can find the treasure at a particular location.

C = Clue, T = Treasure, P = puzzle

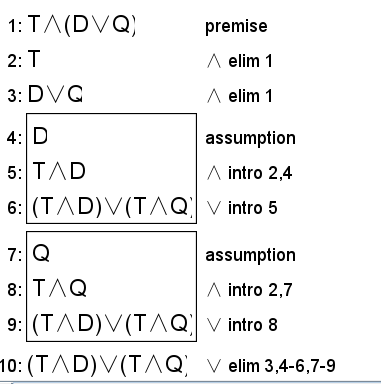


If we have an artifact, we can get no hazard, if we have no hazard and we have solved the puzzle(True) and then we can get the treasure. So we can conclude that if we have an artifact and solve the puzzle(Puzzle is true) and then we can get the treasure.



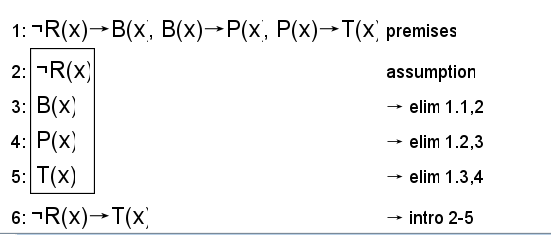
If we have the bridge and river. We have premises as we have a bridge to the clue, we have a true puzzle and no hazard to the clue, and through the clue is true and artifact we can find the treasure. We can conclude that if we have a bridge and puzzle is solved and we have an artifact(True) and we can find the treasure.

T = Treasure; D =Resurrection stones; Q = Marsh



Treasure(T), the object of the game, holds and either the Resurrection stones(D) and Marsh(Q) can alter the process of the game, then the game process can be splitted into two scenarios where Treasure(T) will occur along with Resurrection stones(D) and Marsh(Q).

R = river; B = bridge; P = puzzle; T = treasure



R = River, B = Bride, P = Puzzle, T = Treasure

If we have no river it means we have a bridge, and a bridge can help us find the puzzle, as long as we solved the puzzle(true), we can find the treasure, it means that if there is no river, we can find the treasure anyways.

**Requested Feedback**

To enhance our project, we would appreciate feedback on the following:

**- Model Constraints**: Are there any additional constraints you think would improve the consistency and challenge of the game, particularly around item interactions or environmental effects?

**- Proof Structure**: Does the current setup of our Jape proofs effectively demonstrate logical consistency, or are there improvements in the proof sequence that could strengthen our validation?

**- First-Order Extension**: Are there specific predicate logic extensions or universal constraints you suggest to make our model more comprehensive in a larger or more complex game environment?

- **Jape-Prove**: We are struggling with the jape proofs, we didn’t think of anything to prove the logical stuff.

Useful Notation

Feel free to copy/paste the symbols here and remove this section before submitting.

∧

∨

¬

→

∀

∃

Secondary (Underground World) Proposition draft:

1. Monsters
2. Treasure map (obvious direction on map which indicates the region of the treasure)
3. Random people（helping them to get what they need and they will give us the clue of treasure）