Contents

1   Testing TropICAL's InstAL output

The testing method I'm using is to use traces (sequences of external events) as tests. Each trace should test some kind of function of the system and increase in complexity as the number of tests increases.

1.1   Test 1: Simple trope declaration (PASS)

The purpose of this test is to implement a simple version of the "Hero's Journey" trope. The following trace is the sequence of events a set of agents would need to carry out in order to fulfill this trope in the Star Wars universe:

```
observed(startShow,starWars,0).
observed(go(lukeSkywalker, tatooine),starWars,1).
observed(meet(lukeSkywalker, obiWan),starWars,2).
observed(go(lukeSkywalker, space),starWars,3).
observed(go(lukeSkywalker, tatooine),starWars,4).
```

The following TropICAL code provides a story that the above trace can conform to:

```
"The Hero's Journey" is a trope where:
  The Hero is at Home
  Then the Hero meets the Dispatcher
  Then the Hero goes Away
  Then the Hero returns Home
"Star Wars" is a story:
  It contains the "The Hero's Journey" trope
```

```
"Luke Skywalker" is the Hero
"Obi Wan" is the Dispatcher
"Tatooine" is the Home
"Space" is the Away
```

Note that in the above code, the characters (Luke Skywalker, Obi Wan) fulfill their roles (Hero, Dispatcher) globally across the story. It is also possible to give characters scene-local scope, but that will follow in a later test. The same is true of places and quests in a story. All words are case insensitive.

The events described in the trope simply initiate permissions for the corresponding characters to perform the described actions.

The InstAL this code compiles to is in the test1.ial file (provided in a separate document). This is then compiled to AnsProlog and run through the clingo solver with the above trace. The resulting answer set contains:

```
{:permissions (), :obligations (), :fluents ("phase(herosJourney,inactive)" "role(lukeⁱ
{:permissions ("perm(go(lukeSkywalker,tatooine))"), :obligations (), :fluents ("role(lu
{:permissions ("perm(meet(lukeSkywalker,obiWan))"), :obligations (), :fluents ("role(lu
{:permissions ("perm(go(lukeSkywalker,space))"), :obligations (), :fluents ("role(lukeⁱ
{:permissions ("perm(go(lukeSkywalker,tatooine))"), :obligations (), :fluents ("role(lu
```

1.2 Test 2: Adding a simple quest (FAIL)

Quests are a convenient way to add obligations to tropes, by giving characters tasks to complete. To test this, we add a third line to the trace:

```
observed(startShow,starWars,0).
observed(meet(lukeSkywalker, obiWan),starWars,1).
observed(give(obiWan, lukeSkywalker, quest(destroyTheDeathStar)),starWars,2).
observed(go(lukeSkywalker, space),starWars,3).
observed(destroy(lukeSkywalker, deathStar),starWars,4).
observed(go(lukeSkywalker, tatooine),starWars,5).
```

The quest is added to the story description by referencing it from the trope description and describing the obligations it puts on each character:

```
"The Hero's Journey" is a trope where:
  The Hero is at home
  Then the Hero meets the Dispatcher
  Then the Dispatcher gives the Hero a quest
```

```
    Then the Hero leaves Home
    Then the Hero completes the Quest
    Then the Hero returns Home
"Destroy the Death Star" is a quest where:
    The Hero must go to Space
    The Hero must destroy the Death Star
"Star Wars" is a story:
    It contains the "The Hero's Journey" trope
    "Luke Skywalker" is its Hero
    "Obi Wan" is its Dispatcher
    "Tatooine" is its Home
    "Destroy the Death Star" is its Quest
```

UPDATE: After spending some time on this, it turns out that we really need to use a **bridge institution** to cross-generate events from one institution to another.

The reason for this is that rather than having a special syntax for quests, we need to nest institutions inside other institutions.

This requires restructuring everything so that each trope is contained inside its own institution. This is the best way of having nesting institutions.

1.3 Test 3: A scene within a trope (FAIL)

This time, the trace is the same as above, but with an added event: the start of a scene called "Tatooine".

```
observed(startShow,starWars,0).
observed(scene(tatooine, lukeSkywalker),starWars,1).
observed(meet(lukeSkywalker, obiWan),starWars,2).
observed(tell(obiWan, lukeSkywalker, destroy, deathStar),starWars,3).
observed(go(lukeSkywalker, space),starWars,4).
observed(destroy(lukeSkywalker, deathStar),starWars,5).
observed(go(lukeSkywalker, tatooine),starWars,6).
```

The TropicAL story description has been altered so that the character, place and quest instances have scene-level scope:

```
"The Hero's Journey" is a trope where:
    The Hero is at home
    Then the Hero meets the Dispatcher
    Then the Dispatcher gives the Hero a quest
```

```
  Then the Hero leaves Home
  Then the Hero completes the Quest
  Then the Hero returns home
"Tatooine" is a scene:
  "Luke Skywalker" is its Hero
  "Obi Wan" is its Dispatcher
  "Tatooine" is its home
  "Destroy the Death Star" is its quest
"Destroy the Death Star" is a quest where:
  The Hero must go to Space
  The Hero must destroy the Death Star
"Star Wars" is a story:
  It contains the "The Hero's Journey" trope
  It contains the "Tatooine" scene
```

This means that the entire Hero's Journey trope is contained in the "Tatooine" scene in this case. If the story were to have multiple scenes, the trope's events could be spread out between them.

UPDATE: Needs discussion, but since a scene contains multiple tropes it would be best to use Thomas King's multi-layer institutions to govern tropes in scenes. In any case, this test needs to be left until later.

1.4   Test 4: Situations in tropes (PASS)

This is to test the "When X then Y" syntax for tropes. For this, we return to the simplified hero's journed without a quest. This time, our hero gets a lightsaber:

```
observed(startShow,starWars,0).
observed(go(lukeSkywalker, tatooine),starWars,1).
observed(meet(lukeSkywalker, obiWan),starWars,2).
observed(gets(lukeSkywalker, lightSaber),starWars,3).
observed(bring(lukeSkywalker, hanSolo),starWars,4).
observed(go(lukeSkywalker, space),starWars,5).
observed(go(lukeSkywalker, tatooine),starWars,6).
```

For this test, we alter the trope, adding a conditional based on a situation occurring:

```
"The Hero's Journey" is a trope where:
  The Hero is at Home
```

```
    Then the Hero meets the Dispatcher
    When the Hero gets a Weapon:
      The Hero may bring a Friend
      The Hero may go Away
    Then the Hero goes Away
    Then the Hero returns Home
"Star Wars" is a story:
  It contains the "The Hero's Journey" trope
  "Luke Skywalker" is the Hero
  "Obi Wan" is the Dispatcher
  "Tatooine" is the Home
  "Space" is the Away
  "Han Solo" is the Friend
  "Lightsaber" is the Weapon
```

This states that when the hero gets a weapon, they may find friends and leave home. These permissions are granted to the hero once they have obtained a weapon of some kind.

The answer set contains:

```
{:permissions (), :obligations (), :fluents ("phase(herosJourney,inactive)" "role(lukeS
{:permissions ("perm(go(lukeSkywalker,tatooine))"), :obligations (), :fluents ("role(lu
{:permissions ("perm(meet(lukeSkywalker,obiWan))"), :obligations (), :fluents ("role(lu
{:permissions ("perm(go(lukeSkywalker,tatooine))" "perm(go(lukeSkywalker,space))" "per
{:permissions ("perm(go(lukeSkywalker,tatooine))" "perm(go(lukeSkywalker,space))" "per
{:permissions ("perm(go(lukeSkywalker,tatooine))" "perm(go(lukeSkywalker,space))" "per
{:permissions ("perm(go(lukeSkywalker,tatooine))" "perm(bring(lukeSkywalker,hanSolo))")
```

1.5 Test 5: Obligations without deadlines (FAIL)

```
observed(startShow,starWars,0).
observed(meet(lukeSkywalker, obiWan),starWars,1).
observed(gets(lukeSkywalker, lightSaber),starWars,2).
observed(go(lukeSkywalker, space),starWars,2).
observed(go(lukeSkywalker, tatooine),starWars,3).

"The Hero's Journey" is a trope where:
  The Hero is at home
  Then the Hero meets the Dispatcher
  When the Hero gets a Weapon:
    The Hero must leave Home
```

```
  Then the Hero leaves Home
  Then the Hero returns Home
"Star Wars" is a story:
  It contains the "The Hero's Journey" trope
  "Luke Skywalker" is its Hero
  "Obi Wan" is its Dispatcher
  "Tatooine" is its Home
  "Light Saber" is its Weapon
  "Darth Vader" is its Villain
```

UPDATE: After banging my head against a wall for a while, it turns out that obligations require both deadlines and violation events for InstAL to parse them. Solution is either to require an author to specify deadlines and consequences, or put "dummy" deadlines and consequences in where none are specified.

1.6   Test 6: Obligations with deadlines (FAIL)

```
observed(startShow,starWars,0).
observed(meet(lukeSkywalker, obiWan),starWars,1).
observed(gets(lukeSkywalker, lightSaber),starWars,2).
observed(go(lukeSkywalker, space),starWars,2).
observed(go(lukeSkywalker, tatooine),starWars,3).

"The Hero's Journey" is a trope where:
  The Hero is at home
  Then the Hero meets the Dispatcher
  When the Hero gets a Weapon:
    The Hero must leave Home before the Villain comes
  Then the Hero leaves Home
  Then the Hero returns Home
"Star Wars" is a story:
  It contains the "The Hero's Journey" trope
  "Luke Skywalker" is its Hero
  "Obi Wan" is its Dispatcher
  "Tatooine" is its Home
  "Light Saber" is its Weapon
  "Darth Vader" is its Villain
```

UPDATE: See test 5.

1.7 Test 7: Obligations with deadlines and violation events (FAIL)

NOTE: The syntax of this test is subject to change!

In this case, the villain (Darth Vader) can kill the hero (Luke) if the hero has not left home (Tattoine) before the villain arrives:

```
observed(startShow,starWars,0).
observed(meet(lukeSkywalker, obiWan),starWars,1).
observed(gets(lukeSkywalker, lightSaber),starWars,2).
observed(comes(darthVader, tatooine),starWars,3).
observed(kill(darthVader, lukeSkywalker),starWars,4).
```

This is described in the story as follows:

```
"The Hero's Journey" is a trope where:
  The Hero is at home
  Then the Hero meets the Dispatcher
  When the Hero gets a Weapon:
    The Hero must leave Home before the Villain comes
      Otherwise, the Villain may kill the Hero
  Then the Hero leaves Home
  Then the Hero returns Home
"Star Wars" is a story:
  It contains the "The Hero's Journey" trope
  "Luke Skywalker" is its Hero
  "Obi Wan" is its Dispatcher
  "Tatooine" is its Home
  "Light Saber" is its Weapon
  "Darth Vader" is its Villain
```

The answer set is expected to contain:
UPDATE: Still working on this. Very, very close to passing.

1.8 Test 8: Multiple tropes (FAIL)

This test examines the use of multiple tropes containing situations (When X:), obligations, deadlines and consequences.

```
observed(startShow,starWars,0).
observed(meet(lukeSkywalker, obiWan),starWars,1).
```

```
observed(gets(lukeSkywalker, lightSaber),starWars,2).
observed(captures(darthVader, princessLeia),starWars,3).
observed(go(lukeSkywalker, space),starWars,4).
observed(rescue(lukeSkywalker, princessLeia),starWars,5).
observed(go(lukeSkywalker, tatooine),starWars,6).
```

The story now contains two trope descriptions: "The Hero's Journey" and "The Evil Empire":

```
"The Hero's Journey" is a trope where:
  The Hero is at home
  Then the Hero meets the Dispatcher
  When the Hero gets a Weapon:
    The Hero must leave Home before the Villain comes
      Otherwise, the Villain may kill the Hero
  Then the Hero leaves Home
  Then the Hero rescues the Hostage
  Then the Hero returns Home
"The Evil Empire" is a trope where:
  The Villain gets a Hostage
  When the Villain captures a Hostage:
    The Villain may kill the Hostage
  Then the Villain fights the Hero
  Then the Hero kills the Villain
"Star Wars" is a story:
  It contains the "The Hero's Journey" trope
  It contains the "The Evil Empire" trope
  "Luke Skywalker" is its Hero
  "Obi Wan" is its Dispatcher
  "Tatooine" is its Home
  "Light Saber" is its Weapon
  "Darth Vader" is its Villain
  "Princess Leia" is its Hostage
```

UPDATE: I haven't run this one yet, but it should actually pass easily.

```
(answer set to follow)
```

More tests are to follow, but this is enough for me to be getting on with for now!