CSCI B552: Knowledge-Based Computation

Homework 2
Team Project on Gathering and Representing Everyday Knowledge

Group 3: Mark, Karthik, Saha, Purshottam, Saminda
School of Informatics and Computing
Indiana University Bloomington

Februrary 20, 2012

Table of Contents

1 Concepts

The following tables reveal the form of knowledge representation for entities, connections among entities, properties of entities, and activity relations used in the knowledge capture.

1.1 Entities

Person(P)	Represents a person P
Institution(I)	Represents an Institution I
Entity(E)	Represents any physical entity E
Relation(R)	Represents an abstract relation R
Action(A)	Represents an action A
Event(S)	Represents an Event S

1.2 Connections among Entities

Relationship(E1, E2, [R'])	E1 has a set of relations [R'] with E2

1.3 Properties of Entities

ActionsCapableByEntity(E, A, [E'])	Entity E is capable of or has the ability A using set of Entities [E']
EntityAttributes(E, [A'])	Entity E has a set of attributes [A']
PersonOccupation(P, J)	Occupation of person P is J
JobAction(J, A)	Occupation J involves performing action/activity A
Mood(P, M)	Mood of person P is M
Location(E, L)	Entity E is in location L
InstituteHasOccupation(I, J)	Institute I holds occupation J
WorkFor(E1,E2)	Entity E1 works for Entity E2

1.4 Activity Relations

Action(E1, A, [E'], E2)	Entity E1 performs action A using entities [E'] to/for Entity E2
Event(A)	Defines an Event created because of an Action A
AssignAction(E, A, S)	Entity E is assigned to perform an action A on a Event
ActionCompatibleEvent(A, S)	Defines an Action that can be performed in the given Event
NegativeAction(A)	Actions generally considered negative (eg: fight, kill, cry)
PositiveAction(A)	Actions generally considered positive (eg: help, smile)

E = Entity
R = Relation
I = Institute
A = Action/Activity
M = Mood
L = Location
[X] = Array of X
S = Event
None = Represents empty
P = Person
J = Occupation/Job

2 Assertions

1. Person(P) ⇒ Entity(P)
2. Institution(I) ⇒ Entity(I)
3. Relation(R) ⇒ Entity(R)
4. Action(A) ⇒ Entity(A)
5. Event(S) ⇒ Entity(S)
6. Relationship(E1, E2, [R1]) ∧ Relationship(R1, R2, [subset]) ⇒ Relationship(E1, E2, [R1, R2])
7. Relationship(E1, E2, [R1]) ∧ EntityAttributes(R1, [A']) ⇒ Relationship(E1,E2, [R1] + [A'])
8. AssignAction(E, A, S) ∧ ActionsCapableByEntity(E, A, [E']) ⇒ Action(E, A, [E'], S)
9. ActionsCapableByEntity(E, A, [E']) ∧ JobAction(J, A) ⇒ PersonOccupation(E, J)
10. Action(E1, A, [E'], E2) ∧ NegativeAction(A) ⇒ Mood(E2, Sad) ∨ Mood(E2, Angry)
11. Action(E1, A, [E'], E2) ∧ PositiveAction(A) ⇒ Mood(E2, Happy)
12. Action(E1, A, [E'], E2) ∧ NegativeAction(A) ∧ Relationship(E1, E2, [Friend]) ⇒ Mood(E2, Sad) ∨ Mood(E2, Angry)
13. InstituteHasOccupation(I, J) ∧ PersonOccupation(P, J) ⇒ WorksFor(P, I)
14. WorksFor(P, I) ∧ Location(I, L) ⇒ Location(P, L)
15. JobAction(J, A) ∧ Action(E1, A, [E'], E2) ∧ WorksFor(E1, I) ⇒ Action(I, A, [E'], E2)
16. PersonOccupation(P, J) ∧ Entity(J) ∧ Action(J, A, [E'], E2) ⇒ Action(P, A, [E'], E2)
17. Action(E1, A, [E'], E2) ⇒ Relationship(E1, En, [Interact With]) for each entity 1...n in [E']
18. WorksFor(P, I) ∧ Action(P, A, [E'], E2) ∧ PersonOccupation(P, J) ∧ JobAction(J, A) ⇒ Action(I, A, [E'], E2)

3 Data

In this section, the restricted knowledge domain is presented with the knowledge base constructed from that domain represented in logical form. A list of potential questions to ask is generated and from those a subset is selected to be answered using the concepts and assertions generated in the form of reasoning chains.

3.1 Knowledge Domain

Following sentence has been considered from "Robotics" domain as an example in our study.

Spooner is assigned to investigate the apparent suicide of his friend Alfred Lanning, the roboticist who founded the company U.S. Robotics and created his replacement arm.

3.2 Facts Derived from Knowledge Domain

Person(Spooner)
Person(Alfred)
Institution(US Robotics)
Entity(Replacement Arm)

Relationship(Alfred, Spooner, [Friend])
Relationship(Replacement Arm, Spooner, [Part of])

PersonOccupation(Alfred, Founder)

ActionsCapableByEntity(Alfred, Build Robotics,[ ])
ActionsCapableByEntity(Spooner, Investigate,[ ])

Action(Alfred, Build Robotics, [ ], Replacement Arm)
Action(Alfred, Attached , [Replacement Arm], Spooner)
Action(Alfred, Suicide, [ ], Alfred)

Event(Action(Alfred, Suicide, [ ], Alfred))
ActionCompatibleEvent(Investigate, Event(Action(Alfred, Suicide, [ ], Alfred)))
AssignAction(Spooner,Investigate, Event(Action(Alfred, Suicide, [ ], Alfred)))

NegativeAction(Suicide)

3.3 Additional Knowledge-Base Facts

Entity(Robots)
Entity(Security)
Relation(Friend)
Relation(Acquaintance)

Relation(Founder)
Relation(Detective)
Action(Discover Information)

EntityAttributes(Acquaintance, [know])

Relationship(Friend, Acquaintance, [subset])
Relationship(Replacement Arm, Robotics, [subset])

JobAction(Police Officer, Investigate)
JobAction(Detective, Investigate)
JobAction(Private Eye, Investigate)
JobAction(Roboticist, Build Robotics)

Action(Detective, Investigate, [Security], Discover Information)
EntityAttributes(Investigate, [Discover Information])

InstituteHasOccupation(U.S. Robotics, Build Robotics)
PersonOccupation(U.S. Robotics, Manufacture Robots)
PersonOccupation(Robot, Obey Orders From Humans)

Location(U.S. Robotics, U.S. Robotics Head Office)

4 Questions to Ask

1. What is Spooner's occupation?
2. What is Alfred's occupation?
3. Is Spooner happy or sad?
4. Does Spooner know Alfred?
5. If so, how does Spooner know Alfred? [Concepts where Sponner & Alfred are together and they are involved with some Action eg: Action(Alfred, Attach, [Replacement Arm], Spooner)]
6. Why did Alfred commit suicide?
7. What's the progress of Alfred's suicide case?
8. Where did Alfred commit suicide?
9. How did Alfred commit suicide?
10. Why is Spooner investigating the apparent suicide?
11. Why did Alfred found the company U.S. Robotics?
12. Why did Alfred build a replacement arm?
13. Does Spooner have two biological arms?
14. Is Spooner a friend of Alfred's?
15. When did Alfred commit suicide?
16. What is the evidence that supports that this is a suicide?
17. Were there any witnesses to the suicide?
18. Who reported the incident?
19. To whom was the incident reported?
20. When did Alfred build the replacement arm?
21. When did Alfred found the company U.S. Robotics?
22. Does Spooner still uses the same replacement arm that Alfred built for him?
23. Did U.S. Robotics build Spooner's replacement arm?
24. Does U.S. Robotics build robots?
25. Why does Spooner have a replacement arm?
26. How did Spooner meet Alfred?
27. Did Alfred leave a suicide note?
28. Did Alfred have an ongoing dispute with someone?
29. Did Spooner meet with the U.S. Robotics security?

4.1 Questions to Answer

30. What is Spooner's occupation?
31. Does Spooner know Alfred?
32. Did U.S. Robotics build Spooner's replacement arm?
33. Did Spooner meet with the U.S. Robotics security?
34. Is Spooner happy or sad?

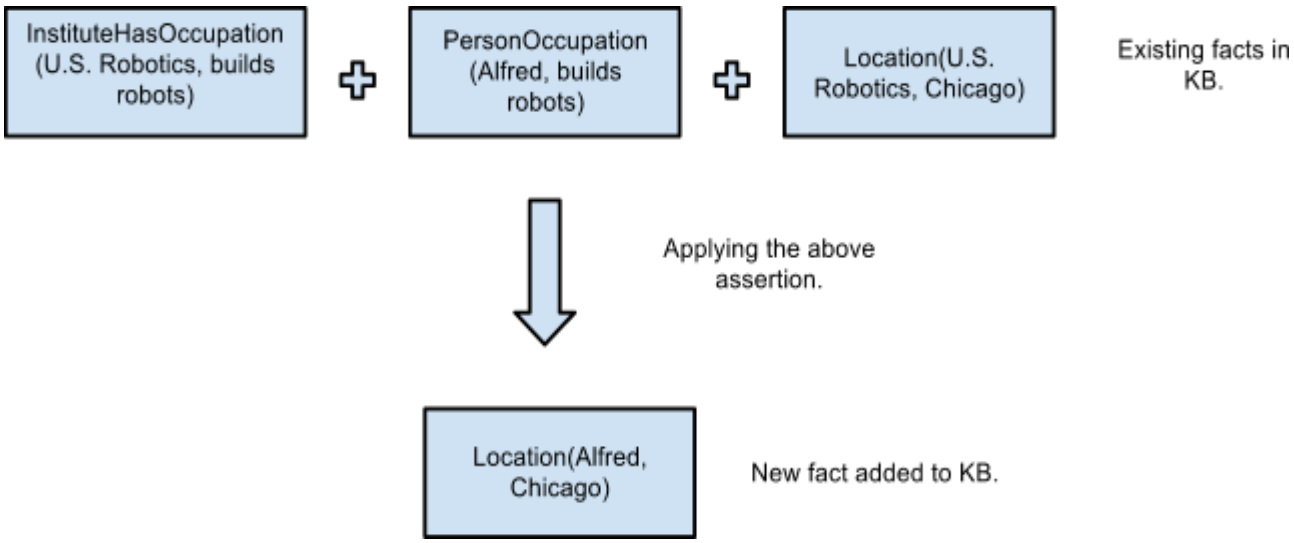
5 Chains of Reasoning

5.1 Applying Rules

To demonstrate how the concepts and assertions chain together to answer a question regarding the knowledge domain, consider the following example. We will be assuming that the information that U.S. Robotics is located in Chicago is available and will be answering the question: Where did Alfred Lanning live?

Assertion:
InstituteHasOccupation(I, J) ⋀ PersonOccupation(P, J) ⋀ Location(I, L) ⇒ Location(P, L)

Applying the rule to the below facts will Infer us the following information.



5.2 Answering the Questions

For the set of questions selected to be answered, the chains of reasoning will be traced with the existing facts and assertions used indicated and the added facts explicitly stated. For the sake of brevity, diagrammatic representations like the one above will be replaced with simple textual traces. Note, that for each question we are stating the new knowledge to be added as if this was the first inference chain encountered by the system, so there will be repeated new facts added across the difference questions.

What is Spooner's occupation?

Existing Knowledge Used:

- Person(Spooner)
- ActionsCapableByEntity(Spooner, Investigate,[])
- JobAction(Detective, Investigate)

Assertions Used:

- Person(P) ⇒ Entity(P)
- ActionsCapableByEntity(E, A, [E']) ∧ JobAction(J, A) ⇒ PersonOccupation(E, J)

Reasoning Chain:

- Person(Spooner) ⇒ Entity(Spooner)
- ActionsCapableByEntity(Spooner, Investigate, []) ∧ JobAction(Detective, Investigate) ⇒ PersonOccupation(Spooner, Detective)

New Knowledge Added:

- Entity(Spooner)
- PersonOccupation(Spooner, Detective)

Answer to Question: **Spooner is a Detective.**

Does Spooner know Alfred?

Existing Knowledge Used:

- Person(Spooner)

Person(Alfred)

- Relation(Friend)
- Relation(Acquaintance)
- Relationship(Alfred, Spooner, [Friend])
- Relationship(Friend, Acquaintance, [subset])
- EntityAttributes(Aquiantance, [knows])

Assertions Used:

- Person(P) ⇒ Entity(P)
- Relation(R) ⇒ Entity(R)
- Relationship(E1, E2, [R1]) ∧ Relationship(R1, R2, [subset])⇒ Relationship(E1, E2, [R2])
- Relationship(E1, E2, [R1]) ∧ EntityAttributes(R1, [A']) ⇒ Relationship(E1,E2, [R1] + [A'])

Reasoning Chain:

- Person(Spooner) ⇒ Entity(Spooner)
- Person(Alfred) ⇒ Entity(Alfred)
- Relation(Friend) ⇒ Entity(Friend)
- Relation(Acquaintance) ⇒ Entity(Acquaintance)
- Relationship(Spooner, Alfred, [Friend]) ∧ Relationship(Friend, Acquaintance, [subset]) ⇒ Relationship(Spooner, Alfred, [Friend, Acquaintance])
- Relationship(Spooner, Alfred, [Friend, Acquaintance]) ∧ EntityAttributes(Acquaintance, [know]) ⇒ Relationship(Spooner, Alfred, [Friend, Acquaintance, know])

New Knowledge Added:

- Entity(Spooner)
- Entity(Alfred)
- Entity(Friend)
- Entity(Acquaintance)
- Relationship(Spooner, Alfred, [Friend, Acquaintance, know])

Answer to Question: **Yes.**

Did U.S. Robotics build Spooner's replacement arm?

Existing Knowledge Used:

- Person(Alfred)
- Entity(Replacement Arm)
- Institution(U.S. Robotics)
- Action(Alfred, Build Robotics, [ ], Replacement Arm)
- ActionsCapableByEntity(Alfred, Build Robotics,[ ])
- JobAction(Roboticist, Build Robotics)
- InstituteHasOccupation(U.S. Robotics, Roboticist)
- Action(Alfred, Build Robotics, [ ], Replacement Arm)

Assertions Used:

- Person(P) ⇒ Entity(P)
- Institution(I) ⇒ Entity(I)
- ActionsCapableByEntity(E, A, [E']) ∧ JobAction(J, A) ⇒ PersonOccupation(E, J)
- InstituteHasOccupation(I, J) ∧ PersonOccupation(P, J) ⇒ WorksFor(P, I)
- WorksFor(P, I) ∧ Action(P, A, [E'], E2) ∧ PersonOccupation(P, J) ∧ JobAction(J, A) ⇒ Action(I, A, [E'], E2)

Reasoning Chain:

- Person(Alfred) ⇒ Entity(Alfred)
- Institute(U.S. Robotics) ⇒ Entity(U.S. Robotics)
- ActionsCapableByEntity(Alfred, Build Robotics, [ ]) ∧ JobAction(Roboticist, Build Robotics) ⇒ PersonOccupation(Alfred, Roboticist)
- InstituteHasOccupation(U.S. Robotics, Roboticist) ∧ PersonOccupation(Alfred, Roboticist) ⇒ WorksFor(Alfred, U.S. Robotics)
- WorksFor(Alfred, U.S. Robotics) ∧ Action(Alfred, Build Robotics, [ ], Replacement Arm) ∧ PersonOccupation(Alfred, Roboticist) ∧ JobAction(Roboticist, Build Robotics) ⇒ Action(U.S. Robotics, Build Robotics, [ ], Replacement Arm)

New Knowledge Added:

- Entity(Alfred)
- Entity(U.S. Robotics)
- PersonOccupation(Alfred, Roboticist)
- WorksFor(Alfred, U.S. Robotics)
- Action(U.S. Robotics, Build Robotics, [ ], Replacement Arm)

Answer to Question: **Yes**.

Did Spooner meet with the U.S. Robotics security?

Existing Knowledge Used:

- Person(Spooner)
- Entity(Security)
- Action(Discover Information)
- Relation(Detective)
- ActionsCapableByEntity(Spooner, Investigate, [ ])
- JobAction(Detective, Investigate)
- Action(Detective, Investigate, [Security], Discover Information)
- EntityAttributes(Investigate, [Discover Information])

Assertions Used:

- Person(P) ⇒ Entity(P)
- Action(A) ⇒ Entity(A)
- Relation(R) ⇒ Entity(R)
- ActionsCapableByEntity(E, A, [E']) ∧ JobAction(J, A) ⇒ PersonOccupation(E, J)
- PersonOccupation(P, J) ∧ Entity(J) ∧ Action(J, A, [E'], E2) ⇒ Action(P, A, [E'], E2)
- Action(E1, A, [E'], E2) ⇒ Relationship(E1, En, [Interact With]) for each entity 1...n in [E']

Reasoning Chain:

- Person(Spooner) ⇒ Entity(Spooner)
- Action(Discover Information) ⇒ Entity(Discover Information)
- Relation(Detective) ⇒ Entity(Detective)
- ActionsCapableByEntity(Spooner, Investigate, [ ]) ∧ JobAction(Detective, Investigate) ⇒ PersonOccupation(Spooner, Detective)
- PersonOccupation(Spooner, Detective) ∧ Entity(Detective) ∧ Action(Detective, Investigate, [Security], Discover Information) ⇒ Action(Spooner, Investigate, [Security], Discover Information)
- Action(Spooner, Investigate, [Security], Discover Information) ⇒ Relationship(Spooner, Security, [Interact With])

New Knowledge Added:

- Entity(Spooner)
- Entity(Discover Information)
- Entity(Detective)
- PersonOccupation(Spooner, Detective)
- Action(Spooner, Investigate, [Security], Discover Information)

- Relationship(Spooner, Security, [Interact With])

Answer to Question: **Yes**.

Is Spooner happy or sad?

Existing Knowledge Used:

- Person(Spooner)
- Person(Alfred)
- Relationship(Alfred, Spooner, [Friend])
- Action(Alfred, Suicide, [ ], Alfred)
- NegativeAction(Suicide)

Assertions Used:

- Person(P) ⇒ Entity(P)
- Action(E1, A, [E'], E2) ∧ NegativeAction(A) ∧ Relationship(E1, E2, [Friend]) ⇒ Mood(E2, Sad) ∨ Mood(E2, Angry)

Reasoning Chain:

- Person(Spooner) ⇒ Entity(Spooner)
- Person(Alfred) ⇒ Entity(Alfred)
- Action(Alfred, Suicide, [ ], Alfred) ∧ NegativeAction(Suicide) ∧ Relationship(Alfred, Spooner, [Friend]) ⇒ Mood(Spooner, Sad) ∨ Mood(Spooner, Angry)

New Knowledge Added:

- Entity(Spooner)
- Entity(Alfred)
- Mood(Spooner, Sad) ∨ Mood(Spooner, Angry)

Answer to Question: **Sad.**

6 Design Decisions and Issues

6.1 Design Overview

The design process for our representation was fairly straight-forward. We elected to start of with high-level entity descriptions (e.g. Person, Institute, etc.) and connect them at the entity level with assertions stating that each of these concepts was also an entity. That then gave us a natural flow into representing entity relations, entity properties, and activity relations since most concepts could be abstracted out to the entity-level.

From there we took an iterative process of building up additional knowledge and assertions that would be needed to answer the selected subset of questions. This process was performed independently for each question with the additional assertions and external knowledge needed added to their respective sets.

What resulted was a set of fairly high-level assertions and commonsense knowledge that when coupled with the facts gleaned from the knowledge domain, could answer questions regarding the knowledge domain where the answer was not explicitly stated.

6.2 Issues Encountered

The major issues we encountered were focused solely in two areas: The entity and assertion abstraction hierarchy and the question generation.

Regarding the question generation, we ran into some conflict where certain questions were posed that one or more of our group members thought would be too difficult or impossible to answer given a small set of commonsense knowledge and the limited facts pulled from the text. In the end, we rephrased the questions to where they may be more easily answered and left them out of the subset of questions to be answered using our representation.

With the entity and assertion abstraction hierarchy, the main issue here was that conceptual representations and the assertions surrounding them were initially too precise. This leads to cases where very similar assertions were needed to deal with different types of concepts. To address this, we reformed the entity hierarchy so that each entity concept could be viewed as an entity at a higher level. This then allowed us to generalize entity-based assertions and eliminated the need for seemingly duplicate assertions.