

#### Exercise 4.64.

Louis Reasoner mistakenly deletes the `outranked-by` rule (section 4.4.1) from the data base. When he realizes this, he quickly reinstalls it. Unfortunately, he makes a slight change in the rule, and types it in as

```
(rule (outranked-by ?staff-person ?boss)
      (or (supervisor ?staff-person ?boss)
          (and (outranked-by ?middle-manager ?boss)
                (supervisor ?staff-person ?middle-manager))))
```

Just after Louis types this information into the system, DeWitt Aull comes by to find out who outranks Ben Bitdiddle. He issues the query

```
(outranked-by (Bitdiddle Ben) ?who)
```

After answering, the system goes into an infinite loop. Explain why.

#### Answer.

Finding that the `outranked-by` rule is applicable, the query system unifies the query

```
(outranked-by (Bitdiddle Ben) ?who)
```

with its conclusion to form an extension on the original empty frame, so `?staff-person` is bound to `(Bitdiddle Ben)` and `?boss` is bound to `?who`. Then, relative to this extended frame, the interpreter proceeds to evaluate the rule body

```
(or (supervisor (Bitdiddle Ben) ?who)
    (and (outranked-by ?middle-manager ?who)
          (supervisor (Bitdiddle Ben) ?middle-manager)))
```

Unifying the `supervisor` rule in the first query of `or` combination extends the frame with the pattern variable `?who` bound to `(Warbucks Oliver)`. However, it is the second one, the `and` of two queries that makes the interpreter falls into infinite loops while making deductions. Note that the interpreter immediately encounters the `outranked-by` rule again as it proceeds to handle the `and` combination. Unifying it extends the stream of frames with `?middle-manager` replacing `(Bitdiddle Ben)` and evaluates almost an identical rule body again

```
(or (supervisor ?middle-manager ?who)
    (and (outranked-by ?middle-manager ?who)
          (supervisor ?middle-manager ?middle-manager)))
```

Observe that the `outranked-by` rule unwind to reveal itself endlessly in the `and` combination with no frame filtered away. Therefore, the query system goes into an infinite loop while handling DeWitt Aull's query.

Going back to the original `outranked-by` rule, which makes the application to `supervisor` rule comes first in the `and` combination, we can see how it avoids the infinite loops.

```
(rule (outranked-by ?staff-person ?boss)
      (or (supervisor ?staff-person ?boss)
          (and (supervisor ?staff-person ?middle-manager)
                (outranked-by ?middle-manager ?boss))))
```

---

\*. Creative Commons  2014, Lawrence X. Amlord (颜世敏, aka 颜序).  
Email address: informlarry@gmail.com

After unification, the query system extends the empty frame with `?staff-person` bound to `(Bitdiddle Ben)` and `?boss` bound to `?who`, it then comes to evaluate the rule body

```
(or (supervisor (Bitdiddle Ben) ?who)
    (and (supervisor (Bitdiddle Ben) ?middle-manager)
          (outranked-by ?middle-manager ?who)))
```

The first query in the `or` combination is unified successfully and extends the frame with the pattern variable `?who` bound to `(Warbucks Oliver)`. Unifying the `supervisor` rule in the `and` combination extends the frame with `?middle-manager` bound to `(Warbucks Oliver)`, then relative to this extended frame, the interpreter reapplies the `outranked-by` rule.

```
(or (supervisor (Warbucks Oliver) ?who)
    (and (supervisor (Warbucks Oliver) ?middle-manager)
          (outranked-by ?middle-manager ?who)))
```

This compound query results in a failed unification and returns an empty frame, for there is no person supervises `(Warbucks Oliver)` according to Microshaft personnel data base. Therefore the unification terminates and the interpreter uses the resulting frame extended by the first query in the `or` combination to instantiate the original query pattern.

```
(outranked-by (Bitdiddle Ben) (Warbucks Oliver))
```