

The Factual Counterfactual Counter

Evante Garza-Licudine, Malvin Gattinger

11.12.2012

Introduction

The MCA framework

Updates

Retraction

Implementation

Modelling

Semantics

Updates

Retraction

Retraction

Examples

Hansson's Hamburger

Cheese and Onion

Conclusion

Results

Future work

The idea

Remember the Hamburger example from homework II.
Could we get a computer to do it for us?

The Factual
Counterfactual
Counter

Evante
Garza-Licudine,
Malvin Gattinger

Introduction

The MCA framework
Updates
Retraction

Implementation

Modelling
Semantics
Updates
Retraction
Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results
Future work

Introduction

The MCA framework
Updates
Retraction

Implementation

Modelling
Semantics
Updates
Retraction
Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results
Future work

The Factual
Counterfactual
Counter

Evante
Garza-Licudine,
Malvin Gattinger

Introduction

The MCA framework
Updates
Retraction

Implementation

Modelling
Semantics
Updates
Retraction
Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results
Future work

Introduction

The Factual
Counterfactual
Counter

Evante
Garza-Licudine,
Malvin Gattinger

Introduction

The MCA framework
Updates
Retraction

Implementation

Modelling
Semantics
Updates
Retraction
Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results
Future work

The MCA framework

The Factual
Counterfactual
Counter

Evante
Garza-Licudine,
Malvin Gattinger

This paper provides an update semantics for counterfactual conditionals. It does so by giving a dynamic twist to the 'Premise Semantics' for counterfactuals developed in Veltman (1976) and Kratzer (1981).

F. Veltman: *Making Counterfactual Assumptions*

Introduction

The MCA framework

Updates

Retraction

Implementation

Modelling

Semantics

Updates

Retraction

Retraction

Examples

Hansson's Hamburger

Cheese and Onion

Conclusion

Results

Future work

The MCA framework

Cognitive States

A cognitive state $S = \langle F_S, U_S \rangle$ is a list of worlds::

	q	p	r
 w_0	0	0	0
w_1	0	0	1
w_2	1	0	0
w_3	1	0	1
w_4	0	1	0
w_5	0	1	1
w_6	1	1	0
w_7	1	1	1

We denote being in F_S by a line | left of the world.
Worlds are not in U_S iff they are stroked out.

Updates

Facts

The Factual
Counterfactual
Counter

Evante
Garza-Licudine,
Malvin Gattinger

We can update a cognitive state with a *fact*/observation:

	q	p	r
w_0	0	0	0
w_1	0	0	1
w_2	1	0	0
w_3	1	0	1
w_4	0	1	0
w_5	0	1	1
w_6	1	1	0
w_7	1	1	1

 $[q \vee \neg r] =$

	q	p	r
w_0	0	0	0
w_1	0	0	1
w_2	1	0	0
w_3	1	0	1
w_4	0	1	0
w_5	0	1	1
w_6	1	1	0
w_7	1	1	1

Updating with a fact only changes F_S .

Introduction

The MCA framework

Updates

Retraction

Implementation

Modelling

Semantics

Updates

Retraction

Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results

Future work

Updates

Laws

The Factual
Counterfactual
Counter

Evante
Garza-Licudine,
Malvin Gattinger

We can update a cognitive state with a *law*:

	q	p	r
w_0	0	0	0
w_1	0	0	1
w_2	1	0	0
w_3	1	0	1
w_4	0	1	0
w_5	0	1	1
w_6	1	1	0
w_7	1	1	1

$$[\Box(p \rightarrow (q \vee r))] =$$

	q	p	r
w_0	0	0	0
w_1	0	0	1
w_2	1	0	0
w_3	1	0	1
w_4	0	1	0
w_5	0	1	1
w_6	1	1	0
w_7	1	1	1

Updating with a law deletes all worlds in which it is false from both F_S and U_S .

Introduction

The MCA framework

Updates

Retraction

Implementation

Modelling

Semantics

Updates

Retraction

Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results

Future work

Retraction

Defining a Basis

Definition 3 (Basis) Let $S = \langle U_S, F_S \rangle$ be a state.

- (i) The situation s *forces* the proposition P within U_S iff for every $w \in U_S$ such that $s \subseteq w$ it holds that $w \in P$.
- (ii) The situation s *determines* the world w iff s forces $\{w\}$ within U_S .
- (iii) The situation s is a *basis for* the world w iff s is a minimal situation determining w within U_S .

Retraction

Retracting Worlds and cognitive states

Definition 4 (Retraction) Let $S = \langle U_S, F_S \rangle$ be a state.

- (i) Suppose $w \in U_S$, and $P \subseteq W$. The set $w \downarrow P$ is determined as follows:
 $s \in w \downarrow P$ iff $s \subseteq w$ and there is a basis s' for w such that s is a maximal subset of s' not forcing P .
- (ii) $S \downarrow P$, the retraction of P from S , is the state $\langle U_{S \downarrow P}, F_{S \downarrow P} \rangle$ determined as follows:
 - (a) $w \in U_{S \downarrow P}$ iff $w \in U_S$
 - (b) $w \in F_{S \downarrow P}$ iff $w \in U_S$ and there are $w' \in F_S$ and $s \in w' \downarrow P$ such that $s \subseteq w$.
- (iii) The state S [if it had been the case that ϕ] is given by $(S \downarrow [\neg\phi])[\phi]$

Implementation

Introduction

The MCA framework
Updates
Retraction

Implementation

Modelling
Semantics
Updates
Retraction
Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results
Future work

- ▶ The list of propositions is given as an argument to construct the neutral cognitive state:

```
1 language=['p', 'q', 'r']  
2 genworlds(language)
```

- ▶ Logical constants:

```
1 phi=" ~(p) "  
2 phi=" (p)&(q) "  
3 phi=" (p) | (q) "  
4 phi=" (p)>(q) "
```

- ▶ Bracket conventions

Modelling

Worls

A world has two sub-structures:

```
1 {  
2   'meta': { 'FS': True, 'US': True, 'name': 'w_3' },  
3   'values': { 'p': 1, 'q': 0, 'r': 1 }  
4 },
```

This one corresponds to this line in a table:

	p	q	r
w_3	1	0	1

Modelling

Cognitive states

The Factual
Counterfactual
Counter

Evante
Garza-Licudine,
Malvin Gatterger

A cognitive state is an array of worlds:

```
1  [
2    { 'meta': {'FS': True, 'US': True, 'name': 'w_0'},
3      'values': {'p': 0, 'q': 0} },
4    { 'meta': {'FS': True, 'US': True, 'name': 'w_1'},
5      'values': {'p': 0, 'q': 1} },
6    { 'meta': {'FS': True, 'US': True, 'name': 'w_2'},
7      'values': {'p': 1, 'q': 0} },
8    { 'meta': {'FS': True, 'US': True, 'name': 'w_3'},
9      'values': {'p': 1, 'q': 1} }
10 ]
```

Introduction

The MCA framework

Updates

Retraction

Implementation

Modelling

Semantics

Updates

Retraction

Retraction

Examples

Hansson's Hamburger

Cheese and Onion

Conclusion

Results

Future work

We use a recursive function to check if a formula is true in a certain world:

```
1 def tiw(world,formula):
2   if len(formula)==1: # atomic
3     return world["values"][formula]
4   else:
5     structure=chop(formula)
6     if structure["connective"]=="~": # negation
7       return not tiw(world,structure["subright"])
8     if structure["connective"]=="&": # conjunction
9       return ( tiw(world,structure["subleft"]) & tiw(world,structure["
10        subright"]) )
11    if structure["connective"]=="|": # disjunction
12      return ( tiw(world,structure["subleft"]) | tiw(world,structure["
13        subright"]) )
14    if structure["connective"]==">": # disjunction
15      return ( tiw(world,structure["subleft"]) <= tiw(world,structure[
16        "subright"]) )
```

Introduction

The MCA framework
Updates
Retraction

Implementation

Modelling
Semantics
Updates
Retraction
Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results
Future work

Updating with a fact

$$S[\phi] = \langle U_S, F_S \cap \llbracket \phi \rrbracket \rangle \text{ if } F_S \cap \llbracket \phi \rrbracket \neq \emptyset;$$
$$S[\phi] = \mathbf{0}, \text{ otherwise.}$$

```
1 def updateFormula(cogstate, formula):
2     newstate = []
3     if formulaIsConsistent(cogstate, formula):
4         for world in cogstate:
5             if not formulaIsTrue(world, formula):
6                 world[meta][FS] = False
7                 newstate.append(world)
8     else:
9         newstate = destroyAllWorlds(cogstate)
10    return newstate
```

Introduction

The MCA framework
Updates
Retraction

Implementation

Modelling
Semantics
Updates
Retraction
Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results
Future work

Updating with a law

The Factual
Counterfactual
Counter

Evante
Garza-Licudine,
Malvin Gattinger

$$S[\Box\phi] = \langle U_S \cap [\Box\phi], F_S \cap [\Box\phi] \rangle \text{ if } F_S \cap [\Box\phi] \neq \emptyset;$$
$$S[\Box\phi] = \mathbf{0}, \text{ otherwise.}$$

```
1 def updateLaw(cogstate, law):
2   newstate = []
3   if formulaIsConsistent(cogstate, law):
4     for world in cogstate:
5       if not formulaIsTrue(world, law):
6         world[meta][FS] = False
7         world[meta][US] = False
8         newstate.append(world)
9   else:
10    newstate = destroyAllWorlds(cogstate)
11  return newstate
```

Introduction

The MCA framework

Updates

Retraction

Implementation

Modelling

Semantics

Updates

Retraction

Retraction

Examples

Hansson's Hamburger

Cheese and Onion

Conclusion

Results

Future work

Retraction

Of a World

We have functions to check if a situation forces, determines or is a basis. Then we can compute the set $w \downarrow P$:

$$w \downarrow P = \{s \subseteq w \mid s \not\subseteq P \wedge \exists s' \text{ basis for } w : s \subset_{\max^*} s'\}$$

```
1 def retractOnWorld(cogstate, worldname, proposition):
2     result=[]
3     world=getWorldByName(worldname, cogstate)
4     for situation in sitgen(world): # s
5         if Forceable(situation, proposition, cogstate):
6             continue # s may not force P
7         adding=False
8         for basis in getAllBases(world, cogstate): # s'
9             if not subset(situation, basis):
10                continue # s is has to be a subset of s'
11                Maximal=True
12                for t in subsitgen(basis):
13                    if Forceable(situation, proposition, cogstate):
14                        continue # t may not force P
15                    if subset(situation, t):
16                        if situation != t:
17                            Maximal=False
18                if not Maximal:
19                    continue # s should be a maximal subset of s'
20                adding=True
21                if adding:
22                    result.append(situation)
23                return result
```

Retraction

Of a State

Retracting a state boils down to retracting all worlds in F_S :

$$U_{S \downarrow P} = U_S$$

$$F_{S \downarrow P} = \{w \in U_S \mid \exists w' \in F_S : \exists s \in w' \downarrow P : s \subseteq w\}.$$

```
1 def retractOnState(cogstate, proposition):
2     result=[]
3     for world in cogstate:
4         newworld={} # do not shoot ourselves in the foot
5         newworld["values"]=dict(world["values"])
6         newworld["meta"]=dict(world["meta"])
7         addingToFS=False
8         if world["meta"]["US"]:
9             for biworld in cogstate:
10                if biworld["meta"]["FS"]:
11                    biretract=retractOnWorld(cogstate, biworld["meta"]["name"],
12                                               proposition)
13                    for s in biretract:
14                        if subset(s, world):
15                            addingToFS=True
16                    newworld["meta"]["FS"]=addingToFS
17                result.append(dict(newworld))
18     return result
```

Retraction

If it had been the case that ϕ

Finally, we can now assume a counterfactual:

```
1 def ifItHadBeenTheCase(cogstate, formula):  
2   # It's so pretty!  
3   return update(retract(cogstate, proposition(cogstate, lnot(formula))  
                  ), formula)
```

This gives us $(S \downarrow [\neg\phi])[\phi]$.

Examples

Introduction

The MCA framework
Updates
Retraction

Implementation

Modelling
Semantics
Updates
Retraction
Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results
Future work

Examples

Hansson's Hamburger

```
1 # Start the tex file
2 out = texheader("Hansson's Hamburger puzzle", "The Factual
   Counterfactual Counter")
3
4 # Need propositional letters for "seeing a man walking with a
   hamburger", "snackbar A is open" and "snackbar B is open".
5 alphabet = ["p", "q", "r"]
6
7 # Now we generate the universe
8 W = worldgen(alphabet)
9 out += texify(W)
10
11 # Update with the fact that we see the man
12 W = updateFormula(W, "r")
13 out += texify(W)
14
15 # Update with the law that if we see a man with a hamburger, he
   must have got it at one of the snackbars
16 W = updateLaw(W, "(r)>((p)|(q))")
17 out += texify(W)
18
19 # Update since we see A is open
20 W = updateFormula(W, "p")
21 out += texify(W)
22
23 # Compute the counterfactual
24 W = ifItHadBeenTheCase(W, "~(p)")
25 out += texify(W)
```

The Factual
Counterfactual
Counter

Evante
Garza-Licudine,
Malvin Gattinger

Introduction

The MCA framework

Updates

Retraction

Implementation

Modelling

Semantics

Updates

Retraction

Retraction

Examples

Hansson's Hamburger

Cheese and Onion

Conclusion

Results

Future work

Examples

Hansson's Hamburger

S_0	q	p	r
w_0	0	0	0
w_1	0	0	1
w_2	1	0	0
w_3	1	0	1
w_4	0	1	0
w_5	0	1	1
w_6	1	1	0
w_7	1	1	1

$$[r] =$$

S_1	q	p	r
w_0	0	0	0
w_1	0	0	1
w_2	1	0	0
w_3	1	0	1
w_4	0	1	0
w_5	0	1	1
w_6	1	1	0
w_7	1	1	1

$$[\Box(r \rightarrow (p \vee q))] =$$

S_2	q	p	r
w_0	0	0	0
w_1	0	0	1
w_2	1	0	0
w_3	1	0	1
w_4	0	1	0
w_5	0	1	1
w_6	1	1	0
w_7	1	1	1

$$S_2[p] =$$

S_3	q	p	r
w_0	0	0	0
w_1	0	0	1
w_2	1	0	0
w_3	1	0	1
w_4	0	1	0
w_5	0	1	1
w_6	1	1	0
w_7	1	1	1

$$\downarrow [\neg\neg p][\neg p] =$$

S_4	q	p	r
w_0	0	0	0
w_1	0	0	1
w_2	1	0	0
w_3	1	0	1
w_4	0	1	0
w_5	0	1	1
w_6	1	1	0
w_7	1	1	1

The last state does not support q , therefore $\neg p \rightsquigarrow q$ is not accepted in S_3 . The same holds for $\neg p \rightsquigarrow r$.

But $\neg p \rightsquigarrow q \vee r$ is accepted.

Cheese and Onion

More propositions, more questions

Increasing the possible worlds increases runtime. How much?

```
1 def checkRandomCounterfactual(cogstate):
2     # generate a random law and update with it:
3     law="("+choice(alphabet)+")>("+choice(alphabet)+")"
4     cogstate = updateLaw(cogstate,law)
5
6     # generate a random fact and update with it:
7     fact=choice(alphabet)
8     cogstate = updateFormula(cogstate ,fact)
9
10    # generate a random non-trivial counterfactual and check it:
11    cfantecedent=choice(alphabet)
12    restralph=list(alphabet)
13    restralph.remove(cfantecedent)
14    cfconsequent=choice(restralph)
15    cogstateNew = ifItHadBeenTheCase(cogstate , cfantecedent)
16    result=supports(cogstateNew ,cfconsequent)
```

Beware: The time needed to check a counterfactual varies.
To get an average result, we ran this function 1000 times on the neutral state for a given number of propositions.

The Factual
Counterfactual
Counter

Evante
Garza-Licudine,
Malvin Gatteringer

Introduction

The MCA framework
Updates
Retraction

Implementation

Modelling
Semantics
Updates
Retraction
Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results
Future work

Conclusion

Introduction

The MCA framework
Updates
Retraction

Implementation

Modelling
Semantics
Updates
Retraction
Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results
Future work

Results

Lessons learned

- ▶ Successfully implemented the semantics from [MCA].
- ▶ Any hamburger-like example can now easily be tried.
- ▶ More than four propositions are hard to cope with.
- ▶ We can now check if interpreting counterfactuals is “just as easy as” interpreting propositional logic ...

The Factual
Counterfactual
Counter

Evante
Garza-Licudine,
Malvin Gattinger

Introduction

The MCA framework
Updates
Retraction

Implementation

Modelling
Semantics
Updates
Retraction
Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results

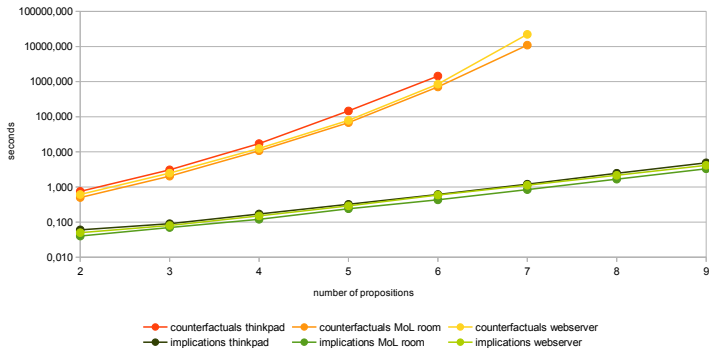
Future work

Results

The Veltman-curve

It is not as easy as material implication.

number of propositions:	2	3	4	5	6	7	8	9
counterfactuals thinkpad	0,750	3,080	17,210	146,380	1442,920			
counterfactuals MoL room	0,500	2,030	10,740	67,970	705,330	10947,400		
counterfactuals webservice	0,610	2,480	12,500	78,760	849,320	22207,320		
implications thinkpad	0,060	0,090	0,170	0,320	0,610	1,200	2,460	4,870
implications MoL room	0,040	0,070	0,120	0,240	0,430	0,840	1,670	3,300
implications webservice	0,050	0,080	0,150	0,290	0,590	1,120	2,150	4,140



The Factual
Counterfactual
Counter

Evante
Garza-Licudine,
Malvin Gattinger

Introduction

The MCA framework

Updates

Retraction

Implementation

Modelling

Semantics

Updates

Retraction

Retraction

Examples

Hansson's Hamburger
Cheese and Onion

Conclusion

Results

Future work

- ▶ Are there further philosophical consequences?
- ▶ What about other counterfactual frameworks?
Can we benchmark against Kratzer, Lewis, ... ?
- ▶ Can the complexity be removed by optimization?
- ▶ What happens in the non-classical case?
Currently we hard-coded:

```
1 truthvalues=[0,1]
```

- ▶ Predicate Logic (This would be hell.)

Got questions? Ask us!

Got counterfactuals? Go to
<http://tinyurl.com/counterfactual>
and check them!

Introduction

The MCA framework

Updates

Retraction

Implementation

Modelling

Semantics

Updates

Retraction

Retraction

Examples

Hansson's Hamburger

Cheese and Onion

Conclusion

Results

Future work