KMM Tutorial 3

1. The purchasing manager for a restaurant performs an analysis of the chef’s favourite dishes and ingredients. She uncovers the following: *Lamb, Beef, Tofu, Squid* and *Prawn* are the most common dishes, cooked in a range of sauces that contain *Cream, CoconutMilk, Chilli, LemonJuice* and *SoySauce*. Letting *cookedIn* and *contains* be binary relations that relate dishes to sauces, and sauces to their ingredients respectively, write necessary and sufficient definitions in the Description Logic ALC for the following:

   i. SweetSauce is a Sauce that contains some *Cream*, and either contains only *Cream*, or contains some *CoconutMilk* and does not contain *Chilli*.

   ii. SourSweetSauce is a Sauce that contains some *CoconutMilk* and contains some *LemonJuice* and contains some *Chilli*.

   iii. HotSauce is a Sauce that contains some *Chilli* and contains only *Chilli* or *SoySauce*.

   iv. RedMeat is *Lamb* or *Beef*.

   v. A WinterDish is RedMeat that is cookedIn only a Sauce that contains some *Chilli*, and is cookedIn some Sauce.

   vi. A SummerDish is *Tofu* or *Squid* that is cookedIn only a Sauce that contains some *CoconutMilk*, and is cookedIn some Sauce.

   vii. Is *Tofu* with SourSweetSauce necessarily a WinterDish? Construct a FACT tableaux to decide whether or not this subsumption relationship holds.

   viii. Is *Tofu* with SourSweetSauce necessarily a SummerDish? Construct a FACT tableaux to decide whether or not this subsumption relationship holds.
vii. Is Tofu with SourSweetSauce necessarily a WinterDish? Construct a FACT tableaux to decide whether or not this subsumption relationship holds. Formally, using FACT:

| Node(a0) | {¬Sauce ⊓ contains.Chilli, Sauce ⊓ contains.CoconutMilk ⊓ contains.LemonJuice ⊓ contains.Chilli} |
| Node(a1) | {¬Sauce, Sauce ⊓ contains.CoconutMilk ⊓ contains.LemonJuice ⊓ contains.Chilli} |
| Node(a1) | {¬Sauce, Sauce ⊓ contains.CoconutMilk ⊓ contains.LemonJuice ⊓ contains.Chilli} |
| add edge labelled cookedIn from a0 to a1 |
| add edge labelled contains from a1 to a2 |

**unless we can show a contradiction between Tofu and ¬RedMeat, disjunct1 remains open**

Continue with the rest of the FACT procedure, although we could stop here.

add edge labelled contains from a1 to a2

disjunct2 shows a clash as disjunct2a and disjunct2b clash
Returning to the open disjunct (disjunct1), there ought to be a way to include the fact that nothing is both Tofu and RedMeat and develop the proof further. Note that the four FACT rules covered in lectures do not allow this goal to be proven.

Disjointness can also be written: \( Tofu \sqcup \neg RedMeat \)

and can be expressed as a constraint: \( Tofu \sqsubseteq \neg RedMeat \) and so \( \{\neg Tofu \sqcup \neg RedMeat\} \) holds universally.

However, this does not lead to a contradiction, just the opposite as if instance a0 instantiates the class Tofu, it necessarily instantiates \( \neg RedMeat \) and so disjunct1 cannot show a clash. Adding \( \neg Tofu \sqcup \neg RedMeat \) to the label set for a0 just returns the proof to the same state, as you can verify.

viii. Is Tofu with SourSweetSauce necessarily a SummerDish? Construct a FACT tableaux to decide whether or not this subsumption relationship holds.

NNF: negation normal form

Node: node in FACT graph