

Alexander hat bei mir die Fragen gestellt; Gerhard hat nur manchmal Tipps gegeben, die haben mir aber mehr geholfen als die Tipps vom Alexander ^^

AL

- Also in der Aussagenlogik, was hatten wir dann da so?
- Gut, dann können wir ja eine Formel beweisen. Wie heißt denn das Kalkül, das wir dafür haben?
- $(C \rightarrow B) \rightarrow (!B \rightarrow !C)$ Was ist denn das? (Ich hab gesagt bestimmt eine Tautologie, aber ich weiß leider nicht welche; da haben sie gelacht und gemeint, ich kann ja mal anfangen das zu beweisen)
- Wie hätte denn der Simplifier das gemacht, hätte der das genauso gemacht? -> Nein, hätte nur die Kalkülregeln die eine Prämisse ergeben angewendet
- Hätte er es denn trotzdem schließen können? -> Ja
- Wie hätte er das gemacht? -> wollten darauf hinaus, dass er kontextgebundene Simplifikation verwendet bei $(C \rightarrow B), C \Rightarrow \dots$ (wusste ich nicht sofort, dass sie auf den Begriff hinaus wollten, haben aber so lange nachgefragt, bis ich kontextabhängige Simplifikation erklärt hatte)

PL

- Wir hatten dann ja Prädikatenlogik, was gilt denn bei der Prädikatenlogik noch, was schon bei der Aussagenlogik galt und was nicht? -> wollten auf Vollständigkeit, Korrektheit, Invertierbarkeit gilt und Entscheidbarkeit gilt nicht mehr hinaus; Entscheidbarkeit hat bei mir ewig gedauert, da haben sie mir sehr viele Tipps gegeben von wegen „gilt mit den Quantoren irgendwas nicht mehr?“, „das kam von Alonzo Church“ kam von Alex und hat mich drauf gebracht ^^ Entscheidbarkeit musste ich dann noch erklären und warum die bei Prädikatenlogik nicht mehr gilt
- Dann wollten sie auf Rewriteregeln und Äquivalenzregeln hinaus; Rewriteregeln musste ich hinschreiben und genauer erklären, wie das funktioniert. Also wonach der Simplifier sucht; also erst nach dem Term t , den er dann durch u ersetzen kann; da wollten sie genauer wissen, wie t in der Sequenz aussehen muss (war am Beispiel $\Rightarrow (m \leq n) \rightarrow (n-m)+m = n$ und da wollten sie drauf hinaus, dass auch „5“ statt dem m dastehen könnte) Dann musste ich noch erklären, wie der Simplifier nach dem ϕ und dem γ suchen würde (ϕ muss im Antezedenz stehen und γ rekursiver Simplifieraufruf hat gereicht)
- Dann sollte ich die für-alle-rechts Regel aus dem Kalkül aufschreiben und dabei erklären -> da wollten sie auch darauf hinaus, warum da y neu sein muss und was passieren würde, wenn es nicht neu wäre

Generiertheit

- Und dann hat uns die Prädikatenlogik aber nicht mehr gereicht, warum war das so? -> Skolem
- Und was haben wir dagegen gemacht? -> Generiertheitsklauseln
- Was genau bewirken denn die Generiertheitsklauseln? (ich hab einfach immer weiter geredet, bis ich an Alex Gesicht gesehen hab, dass ich genau die Sache dazu gesagt hab, die er haben wollte ^^)
- Dann weiß ich nicht mehr, was genau er weiter gefragt hat; es ging aber glaub ich noch ne Weile um Begrifflichkeiten, dann wollten sie auf freie Generiertheit und Data Specifications hinaus und ob man Monomorphie und Korrektheit automatisch hat mit Data Spec.

- Ich sollte auch die Data Specification von listen auf Papier aufschreiben; sie wollten da dann auf die Selektoren hinaus
- Ging dann auch über freie Generiertheit; was da dazu kommt und welche freien Datentypen es denn so gibt (auf Tupel und Aufzählungstypen bin ich nicht gleich gekommen)
- Eindeutigkeit erklären und in welchem Zusammenhang das vorkommt (Anreicherungen)
- Anreicherungen erklären und Hierarchiepersistenz. Alex hat dann immer weiter Fragen gestellt, er wollte unbedingt auf den Begriff „Erweiterung“ hinaus.

KIV

Intersection von zwei Listen strukturell rekursiv spezifizieren

Int-base: $\text{intersection}(x, []) = []$

Int-rec1: $(a \text{ in } x) \rightarrow \text{intersection}(x, a + y) = a + \text{intersection}(x, y)$

Int-rec2: $!(a \text{ in } x) \rightarrow \text{intersection}(x, a + y) = \text{intersection}(x, y)$

und dann auch in KIV ein Lemma dazu beweisen:

$\text{intersection}(\text{append}(x, y), z) = \text{append}(\text{intersection}(x, z), \text{intersection}(y, z))$

1. Sinnvolle Regeln in Simplifier
2. PL Heuristiken an
3. Strukturelle Induktion (erklären, warum über welche Variable!)
4. Jeweils einen der Fälle aus der Rekursion bei den zwei Verzweigungen, die man bekommt anwenden
5. Fertig

Wie immer haben sie echt viele Tipps gegeben und es auch nicht negativ angekreidet, dass man Tipps gebraucht hat, so lange man dann doch noch das gesagt hat worauf sie hinauswollten. Bewertung war auch wieder sehr nett 😊