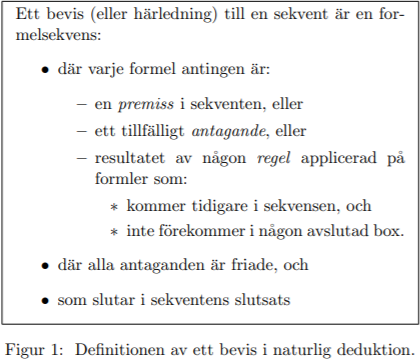
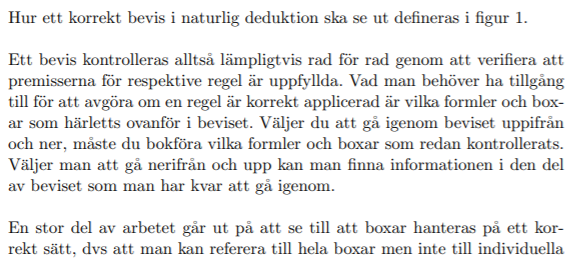
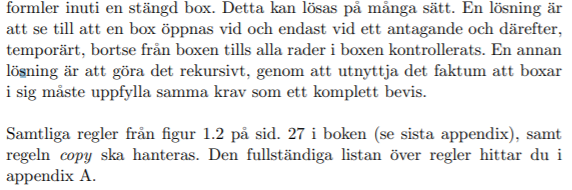
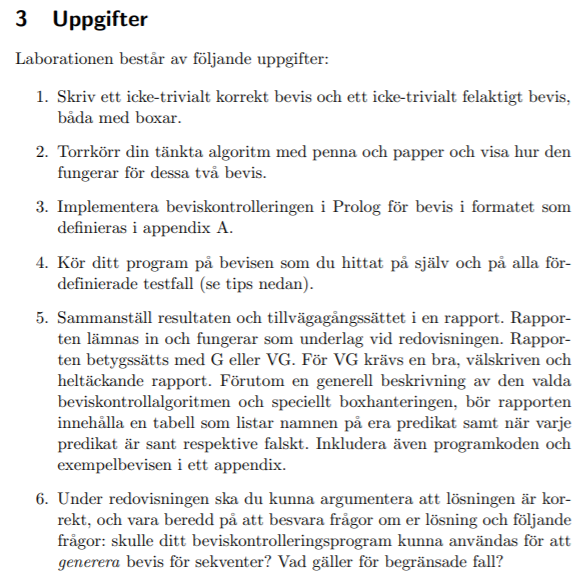
# Lab 2

I den här laborationen ska du konstruera och implementera en algoritm som kontrollerar huruvida ett bevis skrivet i naturlig deduktion (som beskrivet i kursboken) ¨ar korrekt eller inte. Indata till ditt program är en sekvent och ett bevis, och programmet ska svara “yes” om det givna beviset ¨ar korrekt och visar att den givna sekventen gäller och “no” annars.









1.

2.

3.

|  |  |  |
| --- | --- | --- |
| Namn | Parametrar | Predikatet sant/falskt när |
| verify | InputFileName – Filnamnet (inklusive sökväg) på indata. | SANT: Alla tester returnerar sant, annars falskt.  Dvs sant omm vi kan öppna filen (see), läsa in premisser, bevismål och bevis, (stänga filen) och sedan säkerställa att beviset stämmer. |
| valid\_proof | Goal – bevismål  Proof – Listan med bevis | Bevismålet är ekvivalent med slutsatsen i sista raden av Proof, dvs listan med bevis. |
| iterate | 1. Basfall, när listan är tom. 2. Prems – Premisserna för beviset   [Head | Tail] – Listan för beviset, är en recursive metod  Proof – en hel lista med beviset | 1. Sant i samtliga fall då den är ämnad för att hantera basfall. 2. Sant omm ingen iteration av bevislistan returnerar falskt.  Returnerar falskt om fel upptäcks i bevislistan. |
| box\_iterator | 1. Basfall då den givna listan är tom. 2. [BoxHead | BoxTail] – Lista med bevis från inuti boxen. Tail recursive   Proof – En lista med hela beviset | 1. Sant i samtliga fall då den är ämnad för att hantera basfall. 2. Sant omm samtliga tester returnerar sant.   Att ett test returnerar falskt resulterar i även predikatet returnerar falskt. |
| append\_list | 1. [] – Basfall som säkerställer att listan är tom.   List – Den andra listan som skickas in i funktionen.  List – En oinitierad variabel |  |
| find\_nth | N – det radnummer i beviset som skall hittas  Proof – en lista med beviset  Row – Raden som söks |  |
| check\_proof |  |  |
| check\_rule |  |  |
|  |  |  |

4.