1 3a

Anta at 3a er avgjørbart, det vil si at det finnes et program (= maskin = algoritme) P som avgjør om et program Q skriver ut en 'A' eller ikke.

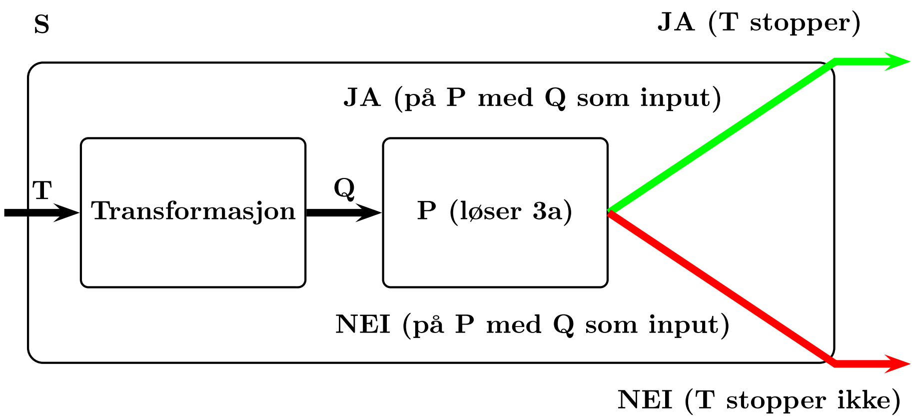
Gitt et program T kan vi konstruere programmet Q på følgende måte:

```
Program Q:
Kjør T
skriv 'A'
```

Vi må også gjøre antagelsen om at T ikke skriver ut 'A'. Dette går greit, da vi kan erstatte alle forekomster av 'A' med et nytt tegn.

Det er klart at Q vil skrive 'A', hvis og bare hvis den kommer til linja hvor det står «skriv 'A'». Altså hvis T stopper.

$$T \text{ stopper } \Leftrightarrow Q \text{ skriver ut 'A'} \Leftrightarrow P \text{ svarer JA}$$



Ved å bruke programmet P på denne måten kan vi konstruere et program S som løser halting-problemet [1]. Siden dette er umulig, må antagelsen vår om at P finnes være gal, og 3a er uavgjørbart.

2 3b

Denne er nesten helt lik forrige oppgave.

Vi antar som i forrige oppgave at et program P finnes som løser 3b, altså at det avgjør om et gitt program Q skriver ut en 'A' hvis input er 'abc' [2].

Gitt et program T konstruerer vi så Q på følgende måte:

[1] Vi løser halting-problemet uten input: Avgjør om T stopper uten input. Dette er like vanskelig som halting-problemet med input, altså uavgjørbart

[2] Vi antar at Q skal skrive ut 'A' hvis *og bare hvis* input er 'abc'. Dersom vi bruker den andre definisjonen, at vi ikke bryr oss om hva som skjer når input ikke er 'abc', kan vi bare bruke *eksakt* samme konstruksjon som i forrige oppgave

```
Program Q: (input = x)
kjør T
if (input == 'abc')
skriv 'A'
```

Her ser vi at den eneste måten Q kan skrive 'A' på, er hvis T stopper *og* input = 'abc'. Dermed kan vi konstruere et program S som løser halting problem som vist i figuren, men der P nå løser 3b, og vi bruker den nye transformasjonen.

Vi får at

$$T \text{ stopper } \Leftrightarrow Q \text{ skriver ut 'A' på input} = \text{'abc'} \Leftrightarrow P \text{ svarer ja.}$$

Med andre ord har vi laget et program som avgjør om et gitt program T stopper, vi har løst halting-problemet, og dermed må antagelsen vår om at P eksisterer være gal, og 3b er også uavgjørbart.

3 3c

Det finnes programmer som kan simulere andre programmer. Dermed kan vi for et gitt program Q med input x finne ut om Q skriver ut en 'A' etter mindre enn 50 skritt, rett og slett ved å simulere Q i 50 skritt. Det kan være at simuleringsprogrammet trenger betydelig mer enn 50 skritt for å gjøre dette, men det spiller ingen rolle, antall skritt er begrenset.

Oppgaven spurte imidlertid om Q ville skrive ut 'A' for et hvilket som helst input. Vi kan ikke teste alle mulige input, det er uendelig mange av dem. Vi observerer imidlertid at et program kan lese inn maks 50 tegn i løpet av 50 skritt. Dermed trenger vi ikke sjekke noe input y med flere enn 50 tegn; Q vil gi eksakt samme resultat med y som med de 50 første tegnene i y — resten av tegnene blir ikke lest.

Antall input med 50 tegn eller mindre er endelig. Det kan være et ekstremt stort tall, mer enn antall atomer i universet hvis vi bruker store og små norske bokstaver, men det er endelig. Med andre ord er det (teoretisk) mulig å teste alle relevante input i endelig tid, og 3c er avgjørbart.

Algoritmen blir da

```
for alle input x < 50 tegn
  simuler Q med x
    hvis Q ikke skriver ut 'A' etter 50 trekk -> return false
return true
```