

## Automatateori

### Lärare

Föreläsningar: Vera Koponen.

Lektioner: Fabian Burghart, Yasmin Tousinejad och Vera Koponen.

### Kurslitteratur

*Formella språk, automater och beräkningar* av Lennart Salling, andra upplagan. Finns tillgänglig i PDF-format på kurssidan på Studium.

### Undervisning

Den lärarledda undervisningen består av 17 föreläsningar och 7 lektioner, vardera om 2×45 minuter. Föreläsningarna kommer att innehålla såväl teori som behandling av exempel och övningsuppgifter. En ungefärlig föreläsningsplan med referenser till kursboken finns nedan. Lektionerna fokuserar på tillämpning av teorin genom problemlösning. Rekommenderade övningsuppgifter till varje lektion finns i separata dokument. *Försök att lösa så många av dessa uppgifter som möjligt inför respektive lektion.* På lektionerna har man möjlighet att få hjälp med uppgifterna av respektive lektionslärare, som även kan välja att gå igenom vissa uppgifter för hela lektionsgruppen om det bedöms som fördelaktigt.

### Examination

Kursen examineras genom en **skriftlig tentamen den 24 oktober 2022**. Glöm inte att anmäla er minst två veckor i förväg.

En **dugga** kommer också att ges den **28 september 2022**. Duggan är inte obligatorisk men *det rekommenderas starkt att ni gör den*, eftersom duggan testar era kunskaper på en del av kursen (mer information nedan). *Dessutom kan ni få vissa uppgifter på tentan tillgodoräknade<sup>1</sup> om ni fått tillräckligt många poäng på duggan.* Glöm ej att anmäla er minst två veckor i förväg till duggan.

### Betygsgränser på tentan och bonus från duggan

Maximalpoängen på *tentan* kommer att vara 40 och för betyg 3, 4 respektive 5 krävs minst 18, 25 respektive 32 poäng. Maximalpoängen på *duggan* kommer att vara 20. Om man får minst 10 respektive minst 15 poäng på duggan så får man de två första, respektive de tre första uppgifterna på tentan tillgodoräknande (så man får full poäng för dem utan att behöva göra dem). *Obs! Det sagda gäller bara till och med den sista omtentan för läsåret 2022/2023.*

---

<sup>1</sup>Att få en uppgift på tentan tillgodoräknad betyder att man får full poäng för den utan att behöva lösa uppgiften. Tanken är att man får denna "frisedel" om man har löst liknande uppgifter korrekt på duggan.

## Ungefärlig föreläsningsplan

Kapitelhänvisningarna syftar på Sallings bok *Formella språk automater och beräkningar*.

Föreläsning	Innehåll	Relevanta kapitel
1	Introduktion, algoritmer, strängar och språk	1.1-1.4
2	Språk, reguljära språk, DFA och NFA	2.1,2.2
3-4	NFA till DFA, reguljärt uttryck till NFA och omvänt	2.3-2.5
5-6	Minimering av en DFA, slutenhetsegenskaper	2.7, 2.6
7	Reguljära språkens gränser	2.8
8	Grammatiker och sammanhangsfria språk (CFL)	3.1-3.2
9	Pushdownautomater (PDA), CFG till PDA	3.3-3.5
10	Sammanhangsfria språkens gränser	3.6-3.7
11	Restriktionsfria grammatiker och eventuellt repetition	4.1
12-13	Turingmaskiner	4.2-4.5
14-15	Universella turingmaskiner och oavgörbarhet	4.6-4.7, 7.1-7.2
16-17	Reservtid och repetition	

## Konkretiserade kursmål som testas på examinationen

1. Veta vad reguljära språk och uttryck är och kunna konstruera sådana för enklare språk.
2. Veta vad NFA och DFA är och kunna konstruera sådana för enklare språk.
3. Omvandla en NFA till DFA (med samma språk) via icke-glupskhet och delmängdsalgoritmen.
4. Omvandla en NFA (eller DFA) till ett reguljärt uttryck med tillståndseliminationsmetoden.
5. Minimering av en DFA med särskiljandealgoritmen.
6. Veta hur NFA:er kan seriekopplas, parallellkopplas och återkopplas, samt känna till och kunna använda slutenhetsegenskaper för reguljära språk.
7. Kunna bevisa att ett språk ej är reguljärt med hjälp av pumpsatsen för reguljära språk eller med särskiljandesatsen och/eller med slutenhetsegenskaper.
8. Veta vad följande begrepp betyder: grammatik, sammanhangsfri grammatik (CFG) samt restriktionsfri grammatik (RFG). Veta vad sammanhangsfria språk (CFL) och restriktionsfria språk (RFL) är.
9. Kunna producera konkreta strängar med hjälp av en grammatik och bestämma om en viss sträng tillhör språket för en viss grammatik. Givet en konkret grammatik, kunna beskriva dess språk (dvs beskriva vilka strängar som kan produceras av den). Kunna konstruera en CFG för en CFL och en RFG för en RFL.

10. Veta vad en PDA är, vad PDA-acceptans innebär och vad en PDA:s språk är.
11. Kunna göra “provkörningar” med en PDA samt avgöra om en konkret sträng tillhör dess språk. Givet en konkret PDA kunna beskriva dess språk. Kunna konstruera en PDA för ett givet språk.
12. Kunna göra en top-down eller bottom-up parser för en sammanhangsfri grammatik.
13. Kunna använda slutenhetsegenskaper för sammanhangsfria språk och kunna visa att ett språk inte är sammanhangsfritt med hjälp av den sammanhangsfria pumpsatsen.
14. Veta vad en Turing-maskin (TM) är och kunna göra provkörningar med en TM på en konkret sträng och se om strängen accepteras, alternativt se vad TM:en ger för output när den har stannat. Givet en konkret TM kunna beskriva vilka strängar som accepteras, alternativt vilka strängar som ger output ‘ja/nej’, alternativt vad den beräknar (om den tex beräknar en funktion). Kunna konstruera en TM som löser en viss uppgift (tex beräknar en funktion eller avgör/accepterar ett språk).
15. Språkhierarkin (s. 182): Veta hur de olika språk-klasserna *Reguljära*, *FA-accepterbara*, *sammanhangsfria*, *PDA-accepterbara*, *TM-avgörbara*, *TM-accepterbara* samt *restriktionsfria* är relaterade till varandra. Som konsekvens av detta ska man kunna svara på frågor som tex “Finns det något språk som är Restriktionsfritt men inte TM-avgörbart”.
16. Kunna svara, med motivation, på frågor om avgörbarhet och accepterbarhet. Det är tillåtet att hänvisa till Church-Turings tes.
17. Bevisa påståenden, möjligen av mer teoretisk/generell natur, om begrepp som behandlats i kursen.

### **Duggan kommer testa följande förmågor**

- Givet en NFA, konstruera, med delmängdskonstruktionen, en DFA som accepterar samma språk.
- Givet en NFA (eller DFA), konstruera med tillståndselimination ett reguljärt uttryck för språket som den accepterar.
- Givet en DFA, konstruera, med särskiljandealgoritmen, en *minimal* DFA som accepterar samma språk.
- Givet ett språk, visa att det är reguljärt genom att konstruera ett en finit automat (DFA eller NFA) eller ett reguljärt uttryck för språket (och eventuellt använda slutenhetsegenskaper), alternativt visa att det inte är reguljärt genom att använda pumpsatsen eller särskiljandesatsen för reguljära språk och eventuellt slutenhetsegenskaper.