Stockholms matematiska cirkel
Datorernas matematik

www.math-stockholm.se/cirkel

16.00–16.15: Fika
16.15–17.15: Föreläsning
17.15–17.30: Rast
17.30–18.00: Gästföreläsning

KTH VETENSKAP OCH KONST
STOCKHOLMS MATEMATIKCENTRUM
STOCKHOLMS UNIVERSITET

Om Cirkeln

- 7 föreläsningar, 7 övningstillfällen
- Övning 6 flyttad från 19 mars till 12 mars
- Schema, program, kartor osv finns på hemsidan:

www.math-stockholm.se/cirkel

Datorernas matematik

1. (19 sep) Vad är matematik, egentligen?
2. (10 okt) Hur kan en dator räkna?
3. (7 nov) Tal med decimaler
4. (12 dec) Tärningen är kastad
5. **(5 feb) Formella språk**
6. (5 mars) Tillståndsmaskiner
7. (26 april) Tillståndsmaskinernas språk

Kapitel 5 – Inledning

- Förra terminen: **hur** kan man räkna med en dator?

Kapitel 5 – Inledning

- Förra terminen: **hur** kan man räkna med en dator?
- Den här terminen: **vad** kan man räkna med en dator?

Kapitel 5 – Inledning

- Förra terminen: **hur** kan man räkna med en dator?
- Den här terminen: **vad** kan man räkna med en dator?

Vi behöver en *modell* av en dator, samt en uppsättning *problem*.

Der Entscheidungsproblem

- Matematiska satser gäller när de kan bevisas ur axiom.

Der Entscheidungsproblem

- Matematiska satser gäller när de kan bevisas ur axiom.
- Fråga 1: kan alla sanna satser härledas ur samma axiomsystem?

Der Entscheidungsproblem

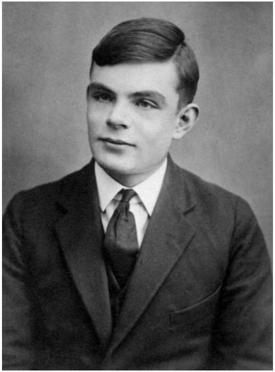
- Matematiska satser gäller när de kan bevisas ur axiom.
- Fråga 1: kan alla sanna satser härledas ur samma axiomsystem?
- Fråga 2: kan bevisprocessen mekaniseras?

Der Entscheidungsproblem

- Matematiska satser gäller när de kan bevisas ur axiom.
- Fråga 1: kan alla sanna satser härledas ur samma axiomsystem?
- Fråga 2: kan bevisprocessen mekaniseras?

Svaret på båda frågorna är nej. Bevisades av Kurt Gödel (fråga 1) och Alan Turing (fråga 2).

Alan Turing (1912-1954)



Filmtips: *The Imitation Game* (2014)

Turingmaskinen

En teoretisk datormaskin som kan simulera alla kända datorer.

Turingmaskinen

En teoretisk datormaskin som kan simulera alla kända datorer.

Vi kommer studera tillståndsmaskiner, en förenklad Turingmaskin. Gås igenom nästa föreläsning!

Beslutsproblem

Först ska vi bestämma vilket problem vi ska lösa. En generell klass är *beslutsproblem*. Låt U vara en mängd.

Beslutsproblem

Först ska vi bestämma vilket problem vi ska lösa. En generell klass är *beslutsproblem*. Låt U vara en mängd.

Givet ett element x i U och en delmängd M i U, ligger x i M?

Beslutsproblem

Först ska vi bestämma vilket problem vi ska lösa. En generell klass är *beslutsproblem*. Låt U vara en mängd.

Givet ett element x i U och en delmängd M i U, ligger x i M?

En lösning på ett beslutsproblem är en metod som tar x och ger svaret 0 eller 1, beroende på om x ligger i M eller ej.

Beslutsproblem

Först ska vi bestämma vilket problem vi ska lösa. En generell klass är *beslutsproblem*. Låt U vara en mängd.

Givet ett element x i U och en delmängd M i U, ligger x i M?

En lösning på ett beslutsproblem är en metod som tar x och ger svaret 0 eller 1, beroende på om x ligger i M eller ej.

Exempel: (tavla)

Kapitel 5.1 – Tecken, alfabet och ord

Nu ska vi specificera vilka element och mängder vi studerar.

Kapitel 5.1 – Tecken, alfabet och ord

Nu ska vi specificera vilka element och mängder vi studerar.

Definition: Ett *alfabet* består av en ändlig mängd *tecken*.

Kapitel 5.1 – Tecken, alfabet och ord

Nu ska vi specificera vilka element och mängder vi studerar.

Definition: Ett *alfabet* består av en ändlig mängd *tecken*.

Godtyckliga tecken betecknas med σ, och alfabet med Σ.

Exempel: (tavla)

Kapitel 5.1 – Tecken, alfabet och ord

Definition: Ett *ord* över ett alfabet är en ändlig sekvens av tecken ur alfabetet.

Kapitel 5.1 – Tecken, alfabet och ord

Definition: Ett *ord* över ett alfabet är en ändlig sekvens av tecken ur alfabetet.

Godtyckliga ord betecknas med w. Ord kallas även *strängar* (eng. *strings*.)

Exempel: (tavla)

Kapitel 5.1 – Tecken, alfabet och ord

Definition: Ett *ord* över ett alfabet är en ändlig sekvens av tecken ur alfabetet.

Godtyckliga ord betecknas med w. Ord kallas även *strängar* (eng. *strings*.)

Exempel: (tavla)

Det *tomma ordet* består av noll tecken, och betecknas ε.

Kapitel 5.1 – Tecken, alfabet och ord

Definition: Ett *ord* över ett alfabet är en ändlig sekvens av tecken ur alfabetet.

Godtyckliga ord betecknas med w. Ord kallas även *strängar* (eng. *strings*.)

Exempel: (tavla)

Det *tomma ordet* består av noll tecken, och betecknas ε.

Definition: Mängden av alla ord över ett alfabet Σ betecknas Σ^*.

Ordoperationer

Definition: *Sammansättningen* av två ord $w = \sigma_1 \cdots \sigma_n$ och $u = \tau_1 \cdots \tau_m$ är ordet $w \circ u = \sigma_1 \cdots \sigma_n \tau_1 \cdots \tau_m$.

Ordoperationer

Definition: *Sammansättningen* av två ord $w = \sigma_1 \cdots \sigma_n$ och $u = \tau_1 \cdots \tau_m$ är ordet $w \circ u = \sigma_1 \cdots \sigma_n \tau_1 \cdots \tau_m$.

Exempel: (tavla)

Ordoperationer

Definition: *Sammansättningen* av två ord $w = \sigma_1 \cdots \sigma_n$ och $u = \tau_1 \cdots \tau_m$ är ordet $w \circ u = \sigma_1 \cdots \sigma_n \tau_1 \cdots \tau_m$.

Exempel: (tavla)

Man skriver inte ut $\circ$, utan skriver wu.

Ordoperationer

Definition: *Sammansättningen* av två ord $w = \sigma_1 \cdots \sigma_n$ och $u = \tau_1 \cdots \tau_m$ är ordet $w \circ u = \sigma_1 \cdots \sigma_n \tau_1 \cdots \tau_m$.

Exempel: (tavla)

Man skriver inte ut $\circ$, utan skriver wu.

Vi definierar $w\varepsilon = \varepsilon w = w$.

Ordoperationer

Definition: *Sammansättningen* av två ord $w = \sigma_1 \cdots \sigma_n$ och $u = \tau_1 \cdots \tau_m$ är ordet $w \circ u = \sigma_1 \cdots \sigma_n \tau_1 \cdots \tau_m$.

Exempel: (tavla)

Man skriver inte ut $\circ$, utan skriver wu.

Vi definierar $w\varepsilon = \varepsilon w = w$.

Obs. Ordningen spelar roll!

Ordoperationer

Definition: Den *n-faldiga upprepningen* av ordet $w = \sigma_1 \cdots \sigma_m$ är ordet

$$w^n = \underbrace{w \circ \cdots \circ w}_{n \text{ stycken}} = \underbrace{\sigma_1 \cdots \sigma_m}_{w} \cdots \underbrace{\sigma_1 \cdots \sigma_m}_{w}.$$

Ordoperationer

Definition: Den *n-faldiga upprepningen* av ordet $w = \sigma_1 \cdots \sigma_m$ är ordet

$$w^n = \underbrace{w \circ \cdots \circ w}_{n \text{ stycken}} = \underbrace{\sigma_1 \cdots \sigma_m}_{w} \cdots \underbrace{\sigma_1 \cdots \sigma_m}_{w}.$$

Exempel: (tavla)

Ordoperationer

Definition: Den *n-faldiga upprepningen* av ordet $w = \sigma_1 \cdots \sigma_m$ är ordet

$$w^n = \underbrace{w \circ \cdots \circ w}_{n \text{ stycken}} = \underbrace{\sigma_1 \cdots \sigma_m}_{w} \cdots \underbrace{\sigma_1 \cdots \sigma_m}_{w}.$$

Exempel: (tavla)

Vi definierar $w^0 = \varepsilon$ för alla ord.

Ordoperationer

Observera följande likhet.

Ord	**Tal**
Upprepning	Potenser
Sammansättning	Multiplikation

Ordoperationer

Observera följande likhet.

Ord	**Tal**
Upprepning	Potenser
Sammansättning	Multiplikation

Sats: För alla ord w och naturliga tal n och m gäller $w^n w^m = w^{n+m}$.

Bevis: (tavla)

Ordoperationer

Observera följande likhet.

Ord	**Tal**
Upprepning	Potenser
Sammansättning	Multiplikation

Sats: För alla ord w och naturliga tal n och m gäller $w^n w^m = w^{n+m}$.

Bevis: (tavla)

Obs. $w^n u^n$ är **inte** lika med $(wu)^n$.

Repetition

Alfabet

Repetition

Alfabet

består av

Repetition

Alfabet

består av

tecken

Repetition

Alfabet

består av

tecken

som bygger upp

Repetition

Alfabet

består av

tecken

som bygger upp

ord

Repetition

Alfabet

består av

tecken

som bygger upp

ord

Ord kan **sammansättas** och **upprepas**.

Repetition

Alfabet

består av

tecken

som bygger upp

ord

Ord kan **sammansättas** och **upprepas**.

Mängden U i beslutsproblemet kommer vara Σ^* för något alfabet. Delmängderna M kommer vara *formella språk*.

Kapitel 5.2 – Formella språk

Definition: Ett *(formellt) språk* över ett alfabet Σ är en mängd av ord över Σ.

Kapitel 5.2 – Formella språk

Definition: Ett *(formellt) språk* över ett alfabet Σ är en mängd av ord över Σ.

Alternativt: ett språk är en delmängd av Σ^*.

Exempel: (tavla)

Kapitel 5.2 – Formella språk

Definition: Ett *(formellt) språk* över ett alfabet Σ är en mängd av ord över Σ.

Alternativt: ett språk är en delmängd av Σ^*.

Exempel: (tavla)

Språk betecknas med bokstaven L (eng. *language*).

Språkoperationer

Eftersom språk är mängder, så kan vi använda vanliga mängdoperationer.

Språkoperationer

Eftersom språk är mängder, så kan vi använda vanliga mängdoperationer.

Definition: *Unionen* av två språk L_1 och L_2 är språket

$$L_1 \cup L_2 = \{w \mid w \in L_1 \text{ eller } w \in L_2\}.$$

Språkoperationer

Eftersom språk är mängder, så kan vi använda vanliga mängdoperationer.

Definition: *Unionen* av två språk L_1 och L_2 är språket

$$L_1 \cup L_2 = \{w \mid w \in L_1 \text{ eller } w \in L_2\}.$$

Exempel: (tavla)

Språkoperationer

En annan språkoperation är denna.

Språkoperationer

En annan språkoperation är denna.

Definition: *Sammansättningen* av två språk L_1 och L_2 är språket

$$L_1 \circ L_2 = \{wu \mid w \in L_1, u \in L_2\}.$$

Språkoperationer

En annan språkoperation är denna.

Definition: *Sammansättningen* av två språk L_1 och L_2 är språket

$$L_1 \circ L_2 = \{wu \mid w \in L_1, u \in L_2\}.$$

Exempel: (tavla)

Språkoperationer

En annan språkoperation är denna.

Definition: *Sammansättningen* av två språk L_1 och L_2 är språket

$$L_1 \circ L_2 = \{wu \mid w \in L_1, u \in L_2\}.$$

Exempel: (tavla)

Obs. $\circ$ skrivs inte ut, så $L_1 \circ L_2 = L_1 L_2$.

Språkoperationer

Precis som för ord så kan man upprepa sammansättningen.

Språkoperationer

Precis som för ord så kan man upprepa sammansättningen.

Definition: Den *n-faldiga upprepningen* av ett språk L är språket

$$L^n = \underbrace{L \circ \cdots \circ L}_{n \text{ stycken}} = \{w_1 \cdots w_n \mid w_1, \ldots w_n \in L\}$$

Språkoperationer

Precis som för ord så kan man upprepa sammansättningen.

Definition: Den *n-faldiga upprepningen* av ett språk L är språket

$$L^n = \underbrace{L \circ \cdots \circ L}_{n \text{ stycken}} = \{w_1 \cdots w_n \mid w_1, \ldots w_n \in L\}$$

Exempel: (tavla)

Språkoperationer

Precis som för ord så kan man upprepa sammansättningen.

Definition: Den *n-faldiga upprepningen* av ett språk L är språket

$$L^n = \underbrace{L \circ \cdots \circ L}_{n \text{ stycken}} = \{w_1 \cdots w_n \mid w_1, \dots w_n \in L\}$$

Exempel: (tavla)

Vi definierar $L^0 = \{\varepsilon\}$.

Språkoperationer

Precis som för ord så kan man upprepa sammansättningen.

Definition: Den *n-faldiga upprepningen* av ett språk L är språket

$$L^n = \underbrace{L \circ \cdots \circ L}_{n \text{ stycken}} = \{w_1 \cdots w_n \mid w_1, \ldots w_n \in L\}$$

Exempel: (tavla)

Vi definierar $L^0 = \{\varepsilon\}$.

Obs. $\{\varepsilon\}$ är **inte** lika med $\emptyset = \{\}$.

Språkoperationer

Den sista språkoperationen vi tittar på är denna.

Språkoperationer

Den sista språkoperationen vi tittar på är denna.

Definition: *Kleenetillslutningen* av ett språk L är språket

$$L^* = L^0 \cup L^1 \cup L^2 \cup \cdots = \bigcup_{n=0}^{\infty} L^n.$$

Språkoperationer

Den sista språkoperationen vi tittar på är denna.

Definition: *Kleenetillslutningen* av ett språk L är språket

$$L^* = L^0 \cup L^1 \cup L^2 \cup \cdots = \bigcup_{n=0}^{\infty} L^n.$$

Exempel: (tavla)

Språkoperationer

Den sista språkoperationen vi tittar på är denna.

Definition: *Kleenetillslutningen* av ett språk L är språket

$$L^* = L^0 \cup L^1 \cup L^2 \cup \cdots = \bigcup_{n=0}^{\infty} L^n.$$

Exempel: (tavla)

Obs. Eftersom $L^0 = \{\varepsilon\}$, så gäller $\varepsilon \in L^*$ för alla språk.

Repetition

Språk

Repetition

Språk

består av

Repetition

Språk

består av

ord

Repetition

Språk

består av

ord

över ett

Repetition

Språk

består av

ord

över ett

alfabet

Repetition

Språk

består av

ord

över ett

alfabet

Man kan beräkna **unionen**, **sammansättningen** och **Kleenetillslutningen** av ett språk.

Kapitel 5.3 – Reguljära språk

Definition: De *reguljära språken* över ett alfabet ges av följande rekursiva definition.

Kapitel 5.3 – Reguljära språk

Definition: De *reguljära språken* över ett alfabet ges av följande rekursiva definition.

- Det tomma språket $\emptyset$ är reguljärt.

Kapitel 5.3 – Reguljära språk

Definition: De *reguljära språken* över ett alfabet ges av följande rekursiva definition.

- Det tomma språket $\emptyset$ är reguljärt.
- Om σ är ett tecken så är $\{\sigma\}$ reguljärt.

Kapitel 5.3 – Reguljära språk

Definition: De *reguljära språken* över ett alfabet ges av följande rekursiva definition.

- Det tomma språket $\emptyset$ är reguljärt.
- Om σ är ett tecken så är $\{\sigma\}$ reguljärt.
- Om L_1 och L_2 är reguljära så är $L_1 \cup L_2$, $L_1 L_2$ och L_1^* reguljära.

Kapitel 5.3 – Reguljära språk

Definition: De *reguljära språken* över ett alfabet ges av följande rekursiva definition.

- Det tomma språket $\emptyset$ är reguljärt.
- Om σ är ett tecken så är $\{\sigma\}$ reguljärt.
- Om L_1 och L_2 är reguljära så är $L_1 \cup L_2$, $L_1 L_2$ och L_1^* reguljära.

Exempel: (tavla)

Kapitel 5.3 – Reguljära språk

Definition: De *reguljära språken* över ett alfabet ges av följande rekursiva definition.

- Det tomma språket $\emptyset$ är reguljärt.
- Om σ är ett tecken så är $\{\sigma\}$ reguljärt.
- Om L_1 och L_2 är reguljära så är $L_1 \cup L_2$, $L_1 L_2$ och L_1^* reguljära.

Exempel: (tavla)

Bindningsregler

För att undvika onödiga parenteser, används bindningsregler.

Bindningsregler

För att undvika onödiga parenteser, används bindningsregler.

1. **Kleenetillslutning**

Bindningsregler

För att undvika onödiga parenteser, används bindningsregler.

1. **Kleenetillslutning**
2. **Sammansättning**

Bindningsregler

För att undvika onödiga parenteser, används bindningsregler.

1. **Kleenetillslutning**
2. **Sammansättning**
3. **Union**.

Bindningsregler

För att undvika onödiga parenteser, används bindningsregler.

1. **Kleenetillslutning**
2. **Sammansättning**
3. **Union**.

Exempel: $\{a\} \cup \{b\}^*\{c\}\{d\} = \{a\} \cup ((\{b\}^*)\{c\}\{d\})$.

Reguljära uttryck

Mängdparenteserna fyller ingen funktion, så de utelämnas. Då får man *reguljära uttryck*.

Reguljära uttryck

Mängdparenteserna fyller ingen funktion, så de utelämnas. Då får man *reguljära uttryck*.

Exempel: $\{a\} \cup \{b\}^*\{c\}\{d\} = a \cup b^*cd$.

Reguljära uttryck

Mängdparenteserna fyller ingen funktion, så de utelämnas. Då får man *reguljära uttryck*.

Exempel: $\{a\} \cup \{b\}^*\{c\}\{d\} = a \cup b^*cd$.

Notera likheten med aritmetiska uttryck.

Reguljära uttryck

Mängdparenteserna fyller ingen funktion, så de utelämnas. Då får man *reguljära uttryck*.

Exempel: $\{a\} \cup \{b\}^*\{c\}\{d\} = a \cup b^*cd$.

Notera likheten med aritmetiska uttryck.

Reguljära språk	**Aritmetiska uttryck**
Kleenetillslutning	Potens
Sammansättning	Multiplikation
Union	Addition

Beslutsproblemet för reguljära språk

Nu kan vi formulera beslutsproblemet rigoröst.

Beslutsproblemet för reguljära språk

Nu kan vi formulera beslutsproblemet rigoröst.

Givet ett ord $w \in \Sigma^*$ *och ett reguljärt språk* $L \subset \Sigma^*$, *ligger* w *i språket* L*?*

Beslutsproblemet för reguljära språk

Nu kan vi formulera beslutsproblemet rigoröst.

Givet ett ord $w \in \Sigma^*$ och ett reguljärt språk $L \subset \Sigma^*$, ligger w i språket L?

De kommande föreläsningarna ska ägnas åt tillståndsmaskiner, och visa att de kan beräkna reguljära språk.

Nästa tillfälle

Nästa gång är om två veckor (20/2). Det är ett övningstillfälle.

Nästa tillfälle

Nästa gång är om två veckor (20/2). Det är ett övningstillfälle.

Nästa föreläsning är om fyra veckor (5/3). Då kommer vi prata om tillståndsmaskiner.