

# SOSI Del 1: SOSI-format notasjon

INNHOLDSFORTEGNELSE

<b>SOSI Del 1: SOSI-format notasjon</b>	<b>1</b>
<b>0 Orientering og introduksjon</b>	<b>3</b>
<b>1 Historikk og status</b>	<b>4</b>
.1.1 Endringslogg fra SOSI-versjon 3.4	4
<b>2 Omfang</b>	<b>5</b>
.2.1 Omfatter	5
.2.2 Målsetting	5
.2.3 Bruksområde	5
<b>3 Normative referanser</b>	<b>6</b>
<b>4 Definisjoner og forkortelser</b>	<b>7</b>
<b>5 SOSI-formatets syntaks</b>	<b>8</b>
.5.1 Notasjon	8
.5.2 Basiselement	10
.5.2.1 Syntaks	11
.5.3 Gruppeelement	11
.5.3.1 Eksempel	11
.5.3.2 Eksempel	12
.5.3.3 Syntaks	12
.5.4 Sammenhengen mellom basiselement og gruppeelement	12
.5.5 Hode	13
.5.5.1 Syntaks	13
.5.6 Brukerdefinerte elementer	13
.5.6.1 Eksempel	13
.5.6.2 Syntaks	14
.5.7 SOSI-formatet	15
.5.7.1 Syntaks	15
.5.8 Konkaterering og kompaktifisering	16
.5.9 Standardelementer	18
.5.10 Konvensjoner	18
.5.11 Syntaks oversikt	19
.5.11.1 Uformelt definert	19
.5.11.2 Innledende definisjoner	19
.5.11.3 Basiselement	19
.5.11.4 Gruppeelement	19
.5.11.5 Hode	19
.5.11.6 Brukerdefinerte elementer	20
.5.11.7 SOSI-formatet	20

---

## 0 Orientering og introduksjon

Denne standarden er en del av SOSI standard familien, og som spesifiserer den syntaktiske oppbyggingen av SOSI-formatet i form av en spesiell form for notasjon (en modifisert versjon av såkalt BNF-notasjon).

Med syntaks menes her den formelle oppbygging av "språket", dvs. reglene for hvordan språkelementene kan stå i forhold til hverandre.

Notasjonen er benyttet for utveksling av data i form av SOSI-filer.

## 1 Historikk og status

Versjon	Dato	Utført av	Grunnlag for endringen
2.0	1992-05		Hovedsaklig kopiert rett fra versjon 1.4
2.21	1996-05		Første revisjon. SOSI-sekr., retting
3.0	1997-07		Definert objekttypedefinisjon
3.1	1999-10		Fast antall desimaler for desimaltall
3.2	2000-05		Ingen endringer
3.3	2001-07		Eksempler knyttet til angivelse av desimaltall med desimaler.
3.4	2002-06		Ingen endring.
4.0	2006-11	SOSI AG 6 / SOSI-sekretariatet	2 nye vderdityper grunnet samordning med internasjonale standarder

Aktuell ansvarlig:

Statens kartverk  
SOSI-sekretariatet  
Kartverksw. 21, 3507 Hønefoss  
Tlf. 32 11 81 00  
[SOSI-sekretariatet@statkart.no](mailto:SOSI-sekretariatet@statkart.no)

### 1.1 Endringslogg fra SOSI-versjon 3.4

Dette dokumentet er en videreføring av SOSI dell\_1 Notasjon i SOSI versjon 3.4.

Innført ny verditype (datatype) Datotid: <datotid>::=<positivt heltall> - på formen ååååmmddttmmss og som er en lovlig dato med presis tidsangivelse

Innført ny verditype (datatype) Boolsk: <boolsk>::=<JA|NEI> - angir om et uttrykk (egenskap) er sann eller falsk

---

## 2 Omfang

### 2.1 Omfatter

---

Spesifikasjon av geografiske data gjøres i form av en implementasjons- og plattformuavhengig modell. Denne modellen må 'mappes' til den plattform hvor utvekslingen av data skal skje.

Norge har i mange år tatt i bruk en egenutviklet syntaks for utveksling av geografiske data, og denne standarden beskriver denne syntaktiske oppbyggingen av SOSI-formatet.

### 2.2 Målsetting

---

En klar spesifisering av SOSI formatets oppbygging sikrer at data kan utveksles i form av SOSI filer mellom ulike brukere og systemer, slik at en oppnår interoperable løsninger.

### 2.3 Bruksområde

---

I utgangspunktet er denne standarden tiltenkt systemleverandører som skal lage eksport/import-rutiner for utveksling av SOSI-filer.

### 3 Normative referanser

Informasjon om BNF (Backus Naur Form) finnes i et uttall bøker og spesifikasjoner. Ulike varianter av BNF og EBNF (extended BNF) har vært introdusert og praktisk talt alle lærebøker om programmeringsspråk har definert sin egen versjon.

For å få til en enhetlig beskrivelse av BNF og spesielt EBNF ble det igangsatt et arbeid i regi av ISO. Dette har resultert i [ISO/IEC 14977:1996\(E\)](#) som definerer en felles og helhetlig beskrivelse av EBNF. Denne standarden er ikke gratis tilgjengelig, mens det siste utkastet ([final draft version \(SC22/N2249\)](#)) er tilgjengelig.

Bruken av BNF i SOSI har imidlertid pågått siden før 1996, og benytter en modifisert versjon av såkalt BNF-notasjon. Følgelig er ikke ISO standarden en normativ referanse for SOSI-Notasjon.

Forøvrig finnes det mye informasjon på nettet vedrørende BNF.

---

## 4 Definisjoner og forkortelser

## 5 SOSI-formatets syntaks

### 5.1 Notasjon

I den syntaktiske ( syntaks: den formelle oppbygging av "språket", dvs. reglene for hvordan språkelementene kan stå i forhold til hverandre ) beskrivelse av SOSI-formatet brukes det en spesiell form for notasjon ( en modifisert versjon av såkalt BNF-notasjon ). Denne skal kort beskrives her. Vi bruker hakker, <...>, til å omslutte navnet på de forskjellige formatelementene. Da formatet er hierarkisk oppbygget, kan et enkelt formatelement representere en ganske komplisert struktur. Eksempelvis sammenfattes den totale beskrivelsen av formatet i et slikt element kalt <SOSI-format>. Verken hakene < og > eller elementnavnet inne i hakene er selv en del av formatet - de hører bare til i beskrivelsen av formatet.

Hvert formatelement defineres i det følgende og symbolet ::= brukes som synonym for 'er definert som'. Altså, hvis det står

```
<abcd> ::= ... beskrivelse ...
```

betyr dette at formatelementet <abcd> er definert som det som står i ... beskrivelse ... .

I selve beskrivelsen, eller definisjonen, vil det forekomme to slags elementer, a) elementer som er definert andre steder i formatbeskrivelsen, og b) såkalte atomer - dette er formatelementer som bare representerer seg selv og de må bli brukt i SOSI-data på nøyaktig samme form som i definisjonen. F. eks. betyr

```
<tekst> ::= "<tegnstreng>"
```

at en tekst er en streng av tegn omsluttet av doble apostrofer "...". Dobbel apostrof,\_"", er da et atom i vår forstand. Et annet eksempel er elementet .SLUTT som avslutter en SOSI-enhet. Dette må stå slik ( altså .SLUTT ) både i beskrivelsen av formatet og i en aktuell forekomst med data. Øvrige tegn som har spesial- betydning er . ( punktum ), som brukes som nivåangivelse, tegnet : ( kolon ), som brukes ved nummerering og referanser, tegnet &, som kan brukes for å skjøte tekster, tegnet @, som brukes for å angi standardverdi og tegnet \* som brukes for å angi en manglende verdi. Tegnene D og E kan brukes som angivelse av eksponent i tall. Alt dette forklares nærmere i det følgende.

Den eneste sammensetningsregelen som finnes, er at elementer kan komme etter hverandre ( i sekvens, jfr. at det er et sekvensielt format ). Det finnes tre måter å beskrive forskjellige måter å danne sekvens på:

En [...] -parentes rundt et eller flere formatelementer betyr at det som står inne i parentes kan utelates. Eks. hvis det står <x> ::= <a>[<b><c>]<d> betyr dette at enten kan <x> bety <a><b><c><d> eller <x> kan bety <a><d>.

En # etter et element betyr at det kan forekomme en eller flere ( vilkårlig antall ) ganger. Eks. hvis det står <x> ::= <a><b># betyr dette at <x> kan stå for <a><b> eller <a><b><b> eller <a><b><b>...<b> der <b> er repetert et visst antall ganger.

En | mellom en sammensetning av elementer betyr at konstruksjonene er alternative. Eks. hvis det står at <x> ::= <a>|<b>|<c> betyr dette at <x> enten kan være <a> eller <x> kan være <b> eller <x> kan være <c>.

Disse tre mulighetene, [...], # og |, kan kombineres på forskjellige måter. F.eks. kan det stå <x> ::= <a><b>|[<a>#]<a>[<b>]<c>#

Da vil alternativ 1 representere <a><b>, alternativ 2 representere mulighetene intet eller tom, <a>, <a><a>, <a><a><a> osv. og alternativ 3 et utall muligheter, f.eks. <a><b><c>, <a><c>, <a><b><c><c><c>, <a><c><c>.



Følgende formatelementer vil bli bare løst beskrevet og ikke formelt definert:

```
<tom> - representerer et tomt element, altså at det ikke opptar  
fysisk plass - nødvendig å ha med for å få enklest mulige  
definisjoner  
<tegn> - alle tegn fra et standard ASCII tegnsett. Lovlige TEGNSETT er  
pr. dato DOSN8, ND7, DECN7, ISO8859-1 og ISO8859-10 (Samisk).  
<nivå 1> - . punktum som angir nivå 1  
<nivå 2> - .. to punktumer, nivå 2  
<nivå n> - ... ialt n punktum , nivå n
```

Vi tar også med en del innledende definisjoner :

```
<siffer>::=0|1|2|3|4|5|6|7|8|9  
  
<positivt heltall>::=<siffer># - ledende nuller er tillatt  
  
<alfanum>::=0|1|2|...|9|A|B|...|Å|a|b|...|å  
  
<heltall>::=[+|-]<positivt heltall>  
  
<tall>::=<heltall>[.<positivt heltall>][E<heltall>]|  
<heltall>[.<positivt heltall>][D<heltall>]  
  
<tegnstreng>::=<tom>|<tegn>#  
  
<tekst>::="<tegnstreng>"|'<tegnstreng>'<tekst>&<tekst>  
  
<standardverdi>::= @ - angir at verdien skal være en tidligere  
definert standardverdi  
  
<dato>::=<positivt heltall> - på formen ååååmmdd og som er en lovlig dato  
  
<datotid>::=<positivt heltall> - på formen ååååmmddttmmss og som er en lovlig dato  
med presis tidsangivelse  
  
<boolsk>::=<JA|NEI> - angir om et uttrykk (egenskap) er sann eller falsk  
  
<serienummer>::=<positivt heltall>:  
  
<referansenummer>::=<heltall>|(:<heltall>)  
  
<spesialverdi>::=<dato>|<referansenummer>  
  
<ikke-verdi>::= * -angir at verdien ikke er gitt i dette tilfellet  
  
<verdi>::=<tall>|<tekst>|<standardverdi>|<ikke-verdi>|<spesialverdi>  
  
<orddeler>::=-|_
```

Skilletegn mellom elementer er nødvendige og det kan være ett eller flere slike. Alle de følgende tegn kan representere skilletegn:

```
blank ASCII-verdi 32  
TAB ASCII-verdi 08  
CR ASCII-verdi 13  
LF ASCII-verdi 10
```

De to siste vil vanligvis forekomme sammen som et par. Skilletegn må forekomme dersom det er nødvendig for å skille to elementer fra hverandre (f.eks. to tall).

Følgende elementer kan ikke deles opp med skilletegn: elementnavn, tallparametre, tekst omsluttet av apostrofer og verdityper.

## 5.2 Basiselement

Den minste informasjonsenhet i SOSI-sammenheng består av et navn som representerer et fysisk objekt, fenomen eller liknende samt dettes tilhørende verdi.

Eksempelvis har en gitt person i Norge et bestemt 11-sifret personnummer, f. eks. 06045049319. Tallet, eller sifrene, i seg selv kan representere hva som helst - et bankkontonummer, en del av et punkts koordinatsett i et eller annet koordinatsystem m.v. Det er først kombinasjonen PERSONNUMMER som beskrivelse og verdien 06045049319 som representerer informasjon. Dette er hovedprinsippet i SOSI-formatet - til enhver gitt verdi skal det henge ved en beskrivelse av hva verdien representerer, og beskrivelse og verdi skal kunne leses i fysisk sammenheng.

Prinsipielt består informasjon på SOSI-format av en sekvens av par - navn og verdi. F. eks. vil en rekke målinger av surhetsgrad på et gitt sted knyttet til dato kunne bli representert slik:

```
.DATO 19830101 ..PH 5.43  
.DATO 19830102 ..PH 5.27  
.DATO 19830103 ..PH 5.63.... osv.
```

I SOSI-sammenheng kaller vi et par bestående av navn og verdi for et basiselement. Dette er det laveste nivå av informasjon. Flere eksempler:

```
GNR 1  
FORNAVN 'ARNE'  
PH 6.17  
BYGN-NR 882909100
```

Elementnavn kan være vilkårlig lange, men bare de 16 første brukes til å skille mellom forskjellige navn. De kan bestå av en kombinasjon av siffer, bokstaver og bindestrek (-) og understreking (\_). Det skilles ikke mellom store og små bokstaver, altså er Gnr og GNR samme elementnavn.

SOSI-formatet opererer med syv forskjellige verdityper eller datatyper - heltall, desimaltall, tekst, dato, datotid, boolsk og referanse. Disse har henholdsvis betegnelsene H, D, T, DATO, DATOTID, BOOLSK og REF, og betegnelsene kan etterfølges av en størrelsesangivelse. Den siste verditypen kalles REF og er et referansenummer, uttrykt som et kolon + tall, f.eks. :100). Nærmere definisjon i kapitlet om brukerdefinerte elementer.

H-verdien brukt alene sier bare at verdien er et positivt eller negativt heltall (fortegnene + og - er lovlige foran selve tallverdien), intet om antall siffer i tallet. Er verditypen angitt til Hn derimot, skal tallverdien ikke overstige n antall siffer.

D-verdien brukt alene sier at tallet er på formen (eventuelt med fortegn) xxxxxx.yyyy og sier intet om antall sifre før eller etter desimalpunktum. Brukes derimot formen Dn skal tallet oppta n eller færre posisjoner totalt. D6.2 betyr tall hvor delen foran punktum angir total lengde, inkludert eventuell fortegn, eksponent, desimaler og desimaltegn, mens det siste tallet står for antall desimaler. Eksponentsialnotasjon på form xxxx.yyEzz er også tillatt. Både E og D aksepteres som eksponentangivelse og begge betyr grunntall 10.

T-verdien brukt alene sier bare at verditypen er tekst, intet om lengden på teksten. Brukes Tn betyr dette at teksten maksimalt består av n tegn, men den kan være kortere.

Sammenhengen mellom et basiselementnavn og dets verditype er enten definert på forhånd (standardelement) eller spesifisert av kilden for dataene (brukerdefinert) i en spesiell definisjonsdel, jfr. kapittel 7.

## 5.2.1 Syntaks

```
<elementnavndel> ::= <orddeler><alfanum>#  
<elementnavn> ::= <alfanum>#<elementnavndel>#  
<basiselement> ::= <elementnavn> <verdi>
```

## 5.3 Gruppeelement

Gruppeelementet er en måte å strukturere sine data på. Det er sjelden en bare trenger å gi uttrykk for en egenskap ved et objekt, som regel er det langt mer kompliserte forhold som må uttrykkes. Gruppeelementet gjør det mulig å bygge opp slike kompliserte strukturer på basis av enklere.

Den struktur en her kan bygge opp kalles en trestruktur, der selve elementet en ønsker seg utgjør rota, andre og enklere grupper utgjør greinene og ytterst fins det bare basiselementer som da utgjør bladene.

### 5.3.1 Eksempel

En person er kjennetegnet ved mange forskjellige egenskaper. Vi kan lage en gruppe av et par av dem på følgende måte :

```
.PERSON  
..NAVN  
...FORNAVN "Per Arne"  
...ETTERNAVN "Hansen"  
..MÅL  
...VEKT 77  
...HØYDE 180
```

Disse dataene er gruppert slik at det er klart at de hører sammen, og er kjennetegn på en og samme forekomst av gruppen PERSON.

Gruppen PERSON er som en ser bygd opp av to likestilte undergrupper, nemlig NAVN og MÅL. Disse er igjen bygd opp av enklere strukturer, i dette tilfellet av basiselementer. Dette er også den generelle måten gruppeelementet bygges opp på.

Et gruppeelement kan gis et serienummer. Dette nummeret må være entydig og brukes til å referere den spesielle forekomsten av gruppen fra et annet sted i samme SOSI-enhet. Derfor finnes det også en måte å referere på.

Eksempel.

```
.BIL_123: ..NUMMER_ 'DD10099' ..FARGE_ 'RØD'
```

er en gruppe som gir en forekomst av BIL. La oss tenke oss at det er en person som eier denne bilen:

```
.PERSON  
..NAVN 'Hans Hansen'  
..BIL  
...NUMMER 'DD10099'  
...FARGE 'RØD'  
da kunne vi ekvivalent ha skrevet  
.PERSON  
..NAVN 'Hans Hansen'  
..BIL :123
```

Vi tenker oss således en slik referanse som en tekstlig substitusjon med de data som referansen peker til. I geodatasammenheng kan et bruksområde være at en har digitalisert en kurve, f. eks. et bekkeløp. I tillegg til å være en bekk er kurven også grense mellom to eiendommer. Det er behov for å referere til kurven både som bekk og fra de to eiendomspolygoner som del av grenseforløpet. Men en ønsker naturligvis ikke å ha representert koordinatmassen tre ganger.

Dette gir en måte å knytte sammen objekter på. Dersom dette ikke er tilstrekkelig, kan en lage ytterligere mekanismer ved å definere egne grupper for dette, f. eks. en gruppe LINK eller KJEDE. Uansett hvilke av disse metodene en bruker, vil det være ens eget ansvar å holde orden på de forekomstene som inngår i slike strukturer.

Som det går fram av syntaks under, kan en referanse ha negativt fortegn. En slik referanse har en helt spesiell tolkning og er bare relevant for grupper som inneholder koordinatverdier. En referanse med negativt fortegn peker til gruppen med tilsvarende positive fortegn, og rekkefølgen på koordinater i denne skal betraktes som tatt i motsatt retning.

### 5.3.2 Eksempel

```
.KURVE 454:  
..PUNKT  
...NORD 8872  
...ØST 410  
..PUNKT  
...NORD 8890  
...ØST 430  
..PUNKT  
...NORD 9020  
...ØST 470
```

Dette representerer en kurve bestående av to rette linjer. Rekkefølgen av koordinatene gir en retning på kurven.

```
.KURVE :-454
```

representerer den samme kurven, men med motsatt retning - altså skrevet ut :

```
.KURVE  
..PUNKT  
...NORD 9020  
...ØST 470  
..PUNKT  
...NORD 8890  
...ØST 430  
..PUNKT  
...NORD 8872  
...ØST 410
```

Et annet spesielt tilfelle er der referansenummeret er satt i parentes, f. eks. (:67). Dette er innført for å løse øy-problematikken, dvs. der en flate forekommer inne i en annen. Inne i parentesen må en referere til en flate, og denne flaten skal trekkes fra hovedflaten. Se også "Del 1-Praktisk bruk" om flater.

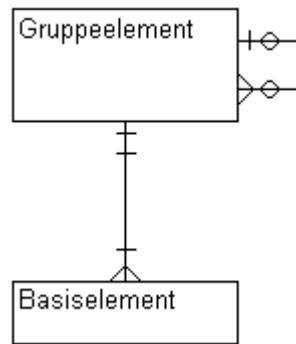
### 5.3.3 Syntaks

```
<serienummer> ::= <positivt heltall>:  
  
<referansenummer> ::= <heltall> | (: <heltall>)  
  
<gruppenelement> ::= <nivå n><basiselement> |  
                  <nivå n><elementnavn>  
                  [<serienummer>]  
                  [<nivå n+1><gruppenelement>]# |  
                  <nivå n><referansenummer>#
```

## 5.4 Sammenhengen mellom basiselement og gruppeelement

Etter forklaringene og eksemplene i de forrige to kapitlene, viser vi grafisk forholdet mellom gruppeelementer og basiselementer i form av en Entitets-Relasjonsmodell (se figur nedenfor). Syntaksen for denne modellen er forklart nærmere i innledningen i Objektkatalogen (Del 2 av denne standarden).

Denne modellen viser at et gruppeelement kan bygges opp av minst ett eller flere basiselementer og null eller flere gruppeelementer. Et basiselement må være en del av et gruppeelement.



Gruppeelementet gjør det mulig å bygge opp slike kompliserte strukturer på basis av enklere, mens basiselement er den minste informasjonsenhet i SOSI-sammenheng. Denne består av et navn som representerer et fysisk objekt, fenomen eller liknende samt dettes tilhørende verdi.

## 5.5 Hode

Det er alltid nødvendig på et eller annet vis å identifisere sine data. En rekke opplysninger kan være aktuelle

- hva er det data om
- hvem har produsert dem
- kilde
- nøyaktighet
- diverse generelle opplysninger
- sammenhenger det er nødvendig å se dataene i
- utvalgsgrupper for å angi spesielle egenskaper for deler av ei SOSI-fil
- osv.

I SOSI-formatet er det tatt med et spesielt element der slike opplysninger kan ligge, dette elementet kaller vi for HODE.

Av form er hodet lik et gruppeelement med et spesielt elementnavn og med den spesielle egenskap at det alltid må ligge først i en SOSI-enhet.

### 5.5.1 Syntaks

```
<hode> ::= .HODE ..<gruppeelement>[ ..<gruppeelement>] #
```

## 5.6 Brukerdefinerte elementer

Som beskrevet tidligere hører det altså med et sett med predefinerte navn, både på basis- og gruppeelementer, til SOSI-formatet. Det er imidlertid helt klart at det ikke er mulig å dekke all behov gjennom disse. Derfor inneholder formatet en mekanisme for å definere nye elementer etter behov - dette vil være elementer som bare er definert innenfor den aktuelle datafil (egentlig SOSI-enhet) og definisjonen vil måtte gjentas ved ønske om å bruke den i andre sammenhenger.

Definisjonen av slike elementer må naturlig nok opptre før de nydefinerte elementnavn brukes. Alle brukerdefinisjoner må samles mellom hode og reelle data.

NB! Dette betyr at hodet utelukkende kan bestå av standardelementer. Til gjengjeld har flere av disse en 'sekkepost'-liknende karakter slik at alle nødvendige opplysninger bør kunne tas med. Brukerdefinisjoner samles i grupper ved bruk av det innledende atom .DEF, definisjonene følger så for hver gruppe eller basiselement på helt analog måte som ved data bortsett fra at verdien nå erstattes av en verditypedefinisjon.

### 5.6.1 Eksempel

Anta at en har bruk for følgende data om en gruppe personer, personnummer, navn, kommunenummer, postadresse. En kunne derfor ønske seg følgende gruppe:

```
.PERSON  
..PERSONNR      - 11 sifret tall  
..NAVN          - f.eks. 30 tegn  
..ADRESSE       - navn på gruppe
```

```
...KOMM      - 4-sifret tall  
...GATEKODE  - 5-sifret tall  
...HUSNR     - tekst  
...POSTNR    - 4-sifret tall
```

Følgende definisjon gjør det mulig å operere med PERSON som gruppenavn :

```
.DEF ! 'Persondata'  
..PERSON      *  
...PERSONNR   H11  
...NAVN       T30  
...ADRESSE    *  
...KOMM       H4  
...GATEKODE   H5  
...HUSNR      T  
...POSTNR     H4
```

Verditypen for gruppenavn er \* - husk at bare basiselement kan ha 'ekte' verditype. Verditypen kan også mangle helt, da må elementnavnet enten være et standardelement eller tidligere definert. En forekomst av PERSON på SOSI-fila kan se slik ut:

```
.PERSON  
..PERSONNR 23037345617  
..NAVN 'HANSEN HANS'  
..ADRESSE  
...KOMM 0219  
...GATEKODE 1466  
...HUSNR "43A"  
...POSTNR 1310
```

Den fulle verdi av muligheten for brukerdefinisjon vil en ikke se før begrepet kompaktifisering blir gjennomgått.

Alle basiselementer må defineres, bl.a. fordi verditypen skal angis. Verditypen står aldri angitt i selve datadelen, altså der en har forekomster av basiselementet. Dersom en trenger basiselementer ut over standardelementene, må en definere dem som under. Det er mulig å angi en standardverdi for elementet i definisjonen - denne må være av samme type som verditypen. Standardverdien vil bli brukt dersom verdien angis som @ i et aktuelt tilfelle av data.

## 5.6.2 Syntaks

```
<verdistørrelse> ::= <tom> | <positivt heltall>  
  
<desimalverdistørrelse> ::= <tom> | <positivt heltall> |  
    <positivt heltall> . <positivt heltall>  
  
<verditype> ::= H <verdistørrelse> |  
    D <desimalverdistørrelse> |  
    T <verdistørrelse> |  
    DATO | svarer til spesialverdien <dato>  
    DATOTID | svarer til spesialverdien <dato>  
    BOOLSK <JA|NEI> |  
    REF | svarer til spesialverdien <referansenummer>  
  
<ikke-verdi> | <tom>  
  
<basisdefinisjon> ::= <elementnavn> <verditype> [<verdi>]  
  
<brukerdefinisjon> ::= <nivå n> <basisdefinisjon> |  
    <nivå n> <elementnavn> <elementnavn> |  
    <nivå n> <elementnavn> *  
    <nivå n+1> <brukerdefinisjon>  
  
<definisjonsgruppe> ::= .DEF <brukerdefinisjon> #
```

```
<definisjonsliste> ::= <definisjonsgruppe>#
```

Objekttyper benytter en egen mekanisme for å beskrive hvilke egenskaper og relasjoner som finnes, og hvilken struktur de har

```
<objekttypenavn> ::= <tegn># -Disse navnene er definert i del 2
```

```
<relasjonstype> ::= S|R S er til geometri, R er til annet
```

```
<relasjonsdefinisjon> ::= <nivå n><elementnavn>  
 <relasjonstype>(<objekttypenavn>)
```

```
<verdiintervall> ::= (<verdi>-<verdi>[ <verdi>-<verdi>]#)
```

```
<verdiliste> ::= (<verdi>[ <verdi>]#)
```

```
<egenskapsdefinisjon> ::= <nivå n><elementnavn> <ikke-verdi>|  
 <verdiintervall>|<verdiliste>
```

```
<egreldef> ::= <egenskapsdefinisjon>|<relasjonsdefinisjon>
```

```
<objekttypedefinisjon> ::= ..OBJTYPE <objekttypenavn>  
 <egreldef>#
```

```
<objekttypedefinisjonsgruppe> ::= ..OBJDEF  
 <objekttypedefinisjon>#
```

```
<objekttypedefinisjonsliste> ::= <objekttypedefinisjonsgruppe>#
```

## 5.7 SOSI-formatet

Vi har nå definert de enkelte delene som SOSI-formatet består av. Det gjenstår å sette det sammen til en helhet.

Kort sagt er SOSI-formatet en eller flere enheter i sekvens som hver representerer en fullverdig informasjonsmengde. Dvs. hver enhet består først av et hode som identifiserer de senere dataene og gir en del informasjon om bakgrunn for dataene og på hvilken måte de skal forstås, f. eks. hvis dataene inneholder koordinater, hvilket koordinatsystem som er brukt. Deretter eventuelt et sett med spesielle brukerdefinisjoner. Så følger selve dataene og formatet avsluttes med det reserverte elementnavnet SLUTT på ytterste nivå.

### 5.7.1 Syntaks

```
<SOSI-data> ::= <gruppeelement>#
```

```
<SOSI-enhet> ::= <hode>  
 <definisjonsliste>  
 <objekttypedefinisjonsliste>  
 <SOSI-data>  
 .SLUTT
```

```
<SOSI-format> ::= <SOSI-enhet>#
```

## 5.8 Konkatenering og kompaktifisering

Dette avsnittet beskriver to viktige mekanismer for å gjøre SOSI-formatet mer fleksibelt.

Konkateneringsmekanismen gjør det mulig å utvide et tidligere definert element med nye egenskaper uten at det er nødvendig med en egen definisjon. (Alle eks. er helt fiktive.)

Anta at en har definert ( formelt upresist ) gruppeelementet PUNKT slik :

```
.PUNKT
..NORD tall
..ØST tall
```

der NORD og ØST representerer koordinatene til PUNKT. Dette kan være definert slik fordi en i en viss sammenheng bare trenger disse to egenskapene ved et punkt i sine data.

Men av og til trenger også å angi et punkts høydeverdi - da er følgende syntaktisk lovlig

```
.PUNKT
..NORD tall
..ØST tall
..HØYDE tall
```

Det samme kunne naturligvis vært oppnådd ved å definere et nytt gruppeelement som

```
.HPUNKT
..NORD tall
..ØST tall
..HØYDE tall
```

Den første metoden har den fordel at en ikke behøver eksplisitt å definere noe nytt gruppeelement for grupper som bare forekommer sjelden. Generelt er det altså lovlig å "hekte" på nye elementer på en eksisterende gruppe. Forutsetningen er selvfølgelig at disse nye elementene også er definert tidligere, enten som standardelement eller ved en brukerdefinisjon. Denne mekanismen strider ikke mot syntaksdefinisjonene. Den andre mekanismen, kompaktifisering eller pakking, er kanskje den viktigste. Ut fra tidligere definisjoner er det jo klart at SOSI-formatet er svært voluminøst i og med at hver dataverdi forutsettes å være forsynt med et tilhørende navn.

Det følgende gir en måte å redusere volumet drastisk på.

```
.PUNKT
..NORD 64000
..ØST 10000
```

Kompaktifiseringsmekanismen gjør følgende form tillatt og ekvivalent til formen over

```
.PUNKT 64000 10000
```

Begrunnelsen for at denne pakking kan tillates er følgende: Fra forekomsten av gruppenavnet PUNKT som enten må være predefinert som standardelement eller eksplisitt definert tidligere på filen, er rekkefølgen av basiselementer (som er de eneste verdibærende) i gruppeelementet kjent. En vet at et PUNKT skal bestå av en NORD(koordinat) etterfulgt av en ØST(koordinat). Dermed er også verdiene 64000 og 10000 entydig tilordnet sine respektive navn. Uansett hierarkisk struktur på en gruppe vil verdirekkefølgen være entydig gitt.



Eksempel (ingen relasjon til objektaktalogen) :

```
.EIENDOM
..G-ID
...KOMM 0220
...GNR 1
...BNR 5
..AREAL
...DYRKET 400
...SKOG 200
...ANNET 600
..SENTRALP
...KOORDSYS 'NGO 1'
...PUNKT
...NORD 32167
...ØST 8456
..REGDATO
...DAG 12
...MND 10
...ÅR 1946
```

kan ekvivalent skrives slik

```
.EIENDOM 0220 1 5 400 200 600 'NGO 1' 32167 8456 12 10 1946
```

som er langt mer økonomisk.

Dersom det etter et gruppenavn følger flere verdier enn gruppedefinisjonen skulle tilsi, repeteres lista.

Eksempel

```
.PUNKT 65199 10111 65201 10120 65300 11001 65305 11134
```

er ekvivalent med

```
.PUNKT 65199 10111 .PUNKT 65201 10120 .PUNKT 65300 11001 .PUNKT 65305 11134
```

som igjen er ekvivalent med

```
.PUNKT ..NORD 65199 ..ØST 10111 .PUNKT ..NORD 65201 ..ØST 10120
.PUNKT ..NORD 65300 ..ØST 11001 .PUNKT ..NORD 65305 ..ØST 11134
```

Det er også lov å blande de to mulighetene over. Dersom eiendommen over omfatter en bygning med gitt bygningsnummer, kan en skrive

```
.EIENDOM 0220 1 5 400 200 600 'NGO 1' 32167 8456 12 10 1946 ..BYGGNR 9413618
```

eller hvis det er flere bygninger

```
.EIENDOM 0220 1 5 400 200 600 'NGO 1' 32167 8456 12 10 1946 ..BYGGNR 9413618 9413626
```

Den måten å gjøre formatet kompakt på som er beskrevet her, består i å 'projisere' gruppedefinisjonen ned på parameterlista. En slik projeksjon kan også gjøres på en avkortet parameterliste, dvs. en liste med færre verdier enn definisjonen skulle tilsi. Regelen er da at verdiene tilordnes de korresponderende basiselementer så langt de rekker, og så avbrytes 'projeksjonen', og en fortsetter med nye elementer.

Eksempel

```
.EIENDOM 0220 1 5 ..BYGGNR 9413618 9413626
```

står for

```
.EIENDOM
..G-ID
...KOMM 0220
```

```
...GNR 1  
...BNR 5  
..BYGGNR 9413618  
..BYGGNR 9413626
```

Altså, 'prosjeksjonen' avbrytes ved første forekomst av eksplisitt angitt basisnavn (..BYGGNR).

## 5.9 Standardelementer

En rekke elementnavn vil være predefinerte - dette gjelder både basiselementer og gruppeelementer. Begrunnelsen for dette er tosidig :

- for å sikre ensartet navn på og beskrivelse av vanlige objekter/fenomener
- for å slippe å definere disse eksplisitt hver gang de ønskes brukt i en SOSI-fil

Organisasjonen med ansvar for vedlikehold av SOSI-formatet fører register over elementnavn med tilhørende verdityper og beskrivelse. Gjeldende standardelementer defineres i Del 2 og oppsummeres i Del 4.

Det er ikke tillatt å omdefinere predefinerte navn i brukerdefinisjoner - konsekvensen av å bryte med dette er implementasjons-avhengig og udefinert her.

## 5.10 Konvensjoner

Kommentarer kan skytes inn hvor som helst på en linje og må begynne med tegnet! . Resten av linja tas som kommentar og ignoreres. Unntatt er selvfølgelig dersom ! forekommer i en tekst, men det må da være en tekst omsluttet av apostrofer (" eller ').

Eldre konverteringssystemer vil kunne ha problemer med å takle SOSI-linjer lenger enn 80 tegn inklusive blanke. Forøvrig er linjelengden ikke relevant, linjeskift er vanlig skilletegn.

Tallparametre kan ikke splittes på to linjer, derimot kan lange tekster fordeles på to eller flere linjer. Teksten må da skrives som to eller flere enkelttekster omsluttet av apostrofer og med tegnet & for å binde dem sammen. Tegnet & kan være skilt fra enkelttekstene med skilletegn.

Dersom det er nødvendig å ha tegnet " eller ' med i teksten, skal det skrives dobbelt, dvs. "" eller "".

Anførselstegn , " eller ' , omkring en tekst kan sløyfes dersom teksten ikke inneholder mellomromstegn ( blank ).

Alle verdier som startet med et punktum må stå i anførselstegn.

Desimaltall kan derfor ikke være på kortformen .5 men må skrives helt ut 0.5.

Eksempelvis vil D6.2 bety desimaltall hvor delen foran punktum angir total lengde, inkludert eventuell fortegn, eksponent, desimaler og desimaltegn, mens det siste tallet står for antall desimaler

Vi har altså følgende muligheter når det gjelder tekst:

```
"" eller ''           - tom tekst  
"hallo, hallo"       - 'normal' tekst : hallo,hallo  
'Peder Aas' ' hus'   - står for : Peder Aas' hus  
'lang tekst ' &     - står for : lang tekst  
'kan vi skrive ' & 'slik'- står for : lang tekst  
                                kan vi skrive slik  
Mjøsa                 - ekvivalent med "Mjøsa"  
                        - eller 'Mjøsa'
```

## 5.11 Syntaks oversikt

### 5.11.1 Uformelt definert

```
<tom> - representer et tomt element,  
        altså at det ikke opptar  
        fysisk plass - nødvendig å ha med for  
        å få enklest mulige definisjoner  
  
<tegn> - alle tegn fra et standard ASCII tegnsett som  
        beskrevet i Del 1, lovlige TEGNSETT er  
        pr. dato DOSN8,ND7,DECN7,ISO8859-1,ISO8859-10  
  
<nivå 1> - .          punktum som angir nivå 1  
<nivå 2> - ..         to punktumer, nivå 2  
<nivå n> - ...        ialt n punktum , nivå n
```

### 5.11.2 Innledende definisjoner

```
<siffer> ::= 0|1|2|3|4|5|6|7|8|9  
  
<positivt heltall> ::= <siffer>#      NB! ledende null tillatt  
  
<heltall> ::= [+|-]<positivt heltall>  
  
<tall> ::= <heltall>[.<positivt heltall>][E<heltall>]|  
          <heltall>[.<positivt heltall>][D<heltall>]  
  
<alfanum> ::= 0|1|2|...|9|A|B|...|Å|a|b|...|å|-|  
  
<tegnstreng> ::= <tom>|<tegn>#  
  
<tekst> ::= "<tegnstreng>" | '<tegnstreng>' | <tekst>&<tekst>  
  
<dato> ::= <positivt heltall> - på formen ååååmmdd  
  
<serienummer> ::= <positivt heltall>:  
  
<referansenummer> ::= <heltall>| (:<heltall>)  
  
<spesialverdi> ::= <dato>|<referansenummer>  
<standardverdi> ::= @  
<ikke-verdi> ::= *  
<verdi> ::= <tall>|<tekst>|<standardverdi>|<ikke-verdi>|<spesialverdi>  
<orddeler> ::= -|_
```

### 5.11.3 Basiselement

```
<elementnavndel> ::= <orddeler><alfanum>#  
<elementnavn> ::= <alfanum>#<elementnavndel>#  
<elementnavn> ::= <alfanum>#  
  
<basiselement> ::= <elementnavn> <verdi>
```

### 5.11.4 Gruppeelement

```
<gruppeelement> ::= <nivå n><basiselement>|  
                  <nivå n><elementnavn> [<serienummer>]  
                  [<nivå n+1><gruppeelement>]#|  
                  <nivå n><referansenummer>#
```

### 5.11.5 Hode

```
<hode> ::= .HODE ..<gruppeelement>[ ..<gruppeelement>]#
```

### 5.11.6 Brukerdefinerte elementer

```
<verdistørrelse> ::= <tom> | <positivt heltall>
<desimalverdistørrelse> ::= <tom> | <positivt heltall> |
<positivt heltall> . <positivt heltall>

<verditype> ::= H <verdistørrelse> |
               D <desimalverdistørrelse> |
               T <verdistørrelse> |
               DATO |          svarer til spesialverdien <dato>
               REF |          svarer til spesialverdien <referansenummer>

<ikke-verdi> | <tom>

<basisdefinisjon> ::= <elementnavn> <verditype> [ <verdi> ]

<brukerdefinisjon> ::= <nivå n> <basisdefinisjon> |
                    <nivå n> <elementnavn> <elementnavn> |
                    <nivå n> <elementnavn> *
                    <nivå n+1> <brukerdefinisjon>

<definisjonsgruppe> ::= .DEF <brukerdefinisjon> #

<definisjonsliste> ::= <definisjonsgruppe> #

<objekttypenavn> ::= <tegn> #

<relasjonstype> ::= S | R

<relasjonsdefinisjon> ::= <nivå n> <elementnavn>
                        <relasjonstype> ( <objekttypenavn> )

<verdiintervall> ::= ( <verdi> - <verdi> [ <verdi> - <verdi> ] # )

<verdiliste> ::= ( <verdi> [ <verdi> ] # )

<egenskapsdefinisjon> ::= <nivå n> <elementnavn> <ikke-verdi> |
                        <verdiintervall> | <verdiliste>

<egreldef> ::= <egenskapsdefinisjon> | <relasjonsdefinisjon>

<objekttypedefinisjon> ::= ..OBJTYPE <objekttypenavn>
<egreldef> #

<objekttypedefinisjonsgruppe> ::= .OBJDEF
                        <objekttypedefinisjon> #

<objekttypedefinisjonsliste> ::= <objekttypedefinisjonsgruppe> #
```

### 5.11.7 SOSI-formatet

```
<SOSI-data> ::= <gruppeelement> #

<SOSI-enhet> ::= <hode>
                [ <definisjonsliste> ]
                [ <objekttypedefinisjonsliste> ]
                <SOSI-data>
                .SLUTT

<SOSI-format> ::= <SOSI-enhet> #
```