

Eksamen 2007 - 2b

Det følgende er et fragment (dvs at ikke-interessante deler av grammatikken ikke er tatt med) av en grammatikk for dette språket.

```
decls → decls ; decl | decl
decl  → var-decl | function-decl
var-decl → type id = expression
function-decl → type id ( parameter? ) body
type → int | bool | void
parameter → type id | type func id
call → id ( id? )
```

Ord i *kursiv* er ikke-terminaler, ord og tegn i **fet** skrift er terminal-symboler. **id** representerer et navn.

En parameter er enten en verdi overført 'by value' eller en funksjon uten parameter. Den enkle reglen i dette språket er at en funksjon med en 'by value'-parameter bare kan kalles med en variabel som aktuell parameter (altså ikke med et generelt uttrykk), mens en funksjon med en funksjonsparameter bare kan kalles med en aktuell parameter som er en funksjon uten parameter. Typen til den aktuelle parameteren skal i begge tilfelle være samme type som den formelle. En funksjon uten parameter må kalles uten aktuell parameter.

Fyll ut de tomme felter i følgende attributtgrammatikk slik at attributtet *ok* for *call* er *true* hvis kallet er gjort ifølge disse regler, ellers *false*. Besvar oppgaven ved å bruke vedlegg side 8.

Du kan anta at det finnes semantiske regler som legger navn inn i symboltabellen. Du kan også anta at $lookup(id.name).kind$ gir verdien 'var' for en variabel og 'func' for en funksjon, $lookup(id.name).type$ er typen til det som *id.name* er navnet på (funksjon, variabel eller parameter) og at $lookup(id.name).has_parameter$ gir verdien 'yes' eller 'no' for en funksjon (med navnet *id.name*) avhengig av om funksjonen har en parameter eller ikke.

Det er ikke behov for å sjekke om funksjonsnavnet (*id*) i en *call*-setning faktisk er deklarerert (du kan altså anta at det allerede er gjort ved andre mekanismer).

Grammar Rule	Semantic Rule
<i>function-decl</i> → type id () <i>body</i>	<i>function-decl.has_parameter</i> = no
<i>function-decl</i> → type id (<i>parameter</i>) <i>body</i>	<i>function-decl.has_parameter</i> = yes <i>function-decl.param-kind</i> = <i>parameter.kind</i> <i>function-decl.param-type</i> = <i>parameter.type</i>
<i>parameter</i> → type id	
<i>parameter</i> → type func id	
<i>type</i> → int	<i>type.type</i> = integer
<i>type</i> → bool	<i>type.type</i> = boolean
<i>type</i> → void	<i>type.type</i> = void
<i>call</i> → id ()	<i>call.ok</i> = (<i>lookup(id.name).has_parameter=no</i>)
<i>call</i> → id₁ (id₂)	<i>call.ok</i> =