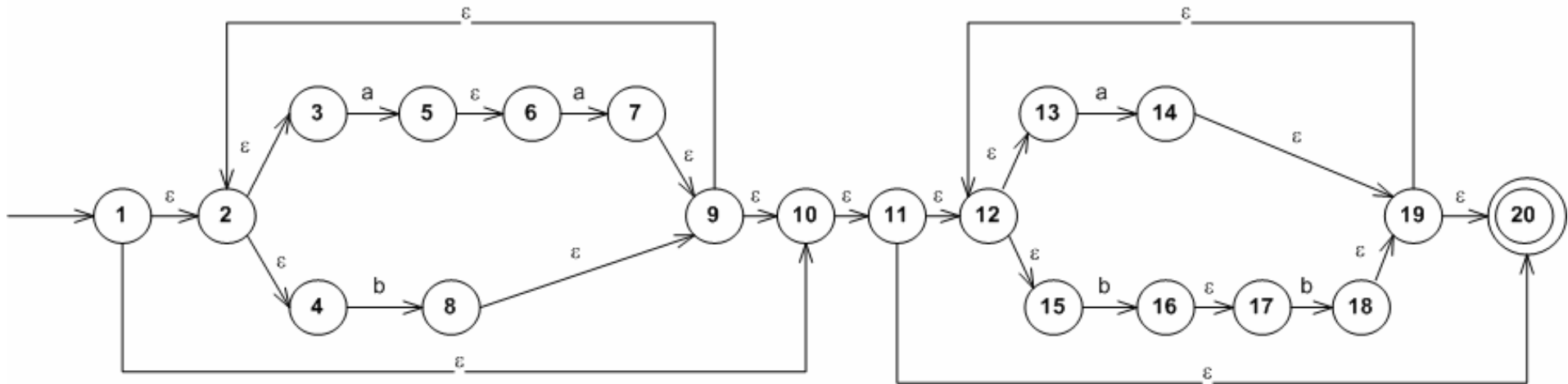


2005 – opppg. 1

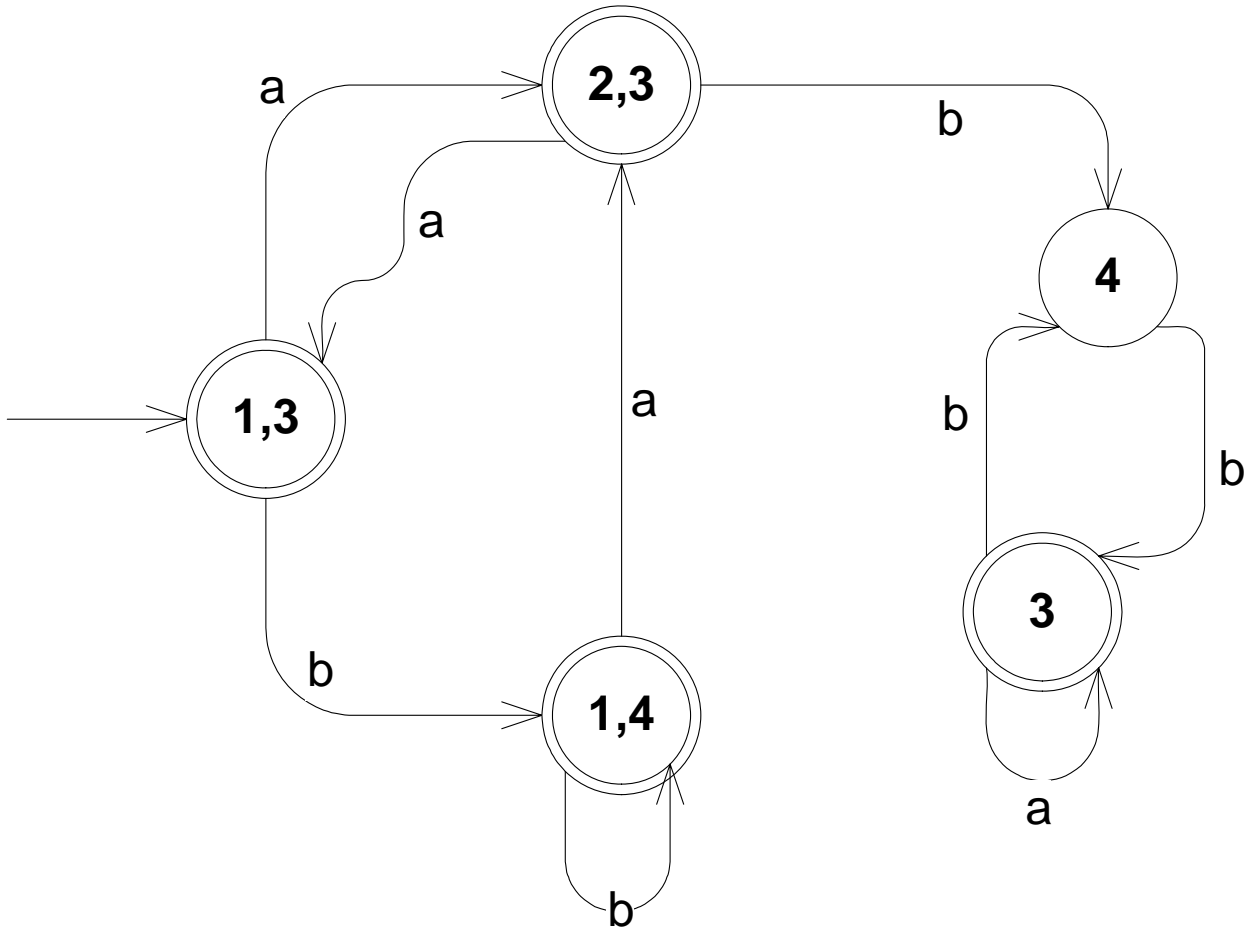
- a)



- b)

$(aa \mid b)^* (a \mid bb)^*$

▪ c)



2005 oppg. 3

- a)

Grammar Rule	Semantic Rule
<code>class → class name { decls }</code>	<code>decls.enclosingClassName = name.name</code>
<code>decls₁ → decls₂ ; decl</code>	<code>decls₂.enclosingClassName = decls₁.enclosingClassName</code> <code>decl.enclosingClassName = decls₁.enclosingClassName</code>
<code>decls → decl</code>	<code>decl.enclosingClassName = decls.enclosingClassName</code>
<code>decl → variable-decl</code>	
<code>decl → method-decl</code>	<code>method-decl.enclosingClassName = decl.enclosingClassName</code>
<code>method-decl → type name (params) body</code>	<code>if (name.name = method-decl.enclosingClassName) then if (not(type.type = void))then error("constructor not of type void")</code> <code>eller</code> <code>if (name.name = method-decl.enclosingClassName) and (not(type.type = void))then error("constructor not of type void")</code>
<code>type → int</code>	<code>type.type = int</code>
<code>type → bool</code>	<code>type.type = bool</code>
<code>type → void</code>	<code>type.type = void</code>

▪ b)

C1-objekt

int i
vt
int j

0
1
2
3

C1P
BQ
BR
C1S

C2-objekt

int i
vt
int j
int k

0
1
2
3
4

AP
BQ
C2R
BS
C2T

- c) Nei: Virtuelle metoder i B-objekter kan fremdeles kaldes via A-typede pekere.
- d) Rett frem løsning uten å bryte ut av løkken når match finnes. Variablen 'instanceof' er av type bool, mens variable 'cd' er en referance som kan referere klassesdeskriptorer.

```
instanceof = false;  
cd = <refExpr>.cl;  
while not(cd = klassesdeskriptor for klassen Object) do  
  {if cd = klassesdeskriptor for klassen <class>  
    then instanceof = true;  
    cd = cd.cl}
```

- e)
 - Generell test:
 - Sjekk at `<class>.subklassenivå` ligger innenfor grensene på den aktuelle supers-tabellen
 - `<refExpr>.vt.cl.supers[<class>.subklassenivå] = <class>` Supers for C11 og C21

– Supers for C11 og C21

0	Object
1	A
2	B
3	C1
4	C11

0	Object
1	A
2	B
3	C2
4	C21

- Konkrete tester:
 - `rc11.vt.cl.supers[3]= C1` dvs `C1 = C1` dvs true
 - `rc11.vt.cl.supers[3]= C2` dvs `C1 = C2` dvs false

2006

1a

Program 1: 0

Program 2: 2

1b

Program 1: 0

Program 2: 2

Grammar Rule

Semantic Rule

■ c)

procedure →

insert(id.name, param.kind)

proc id (*param*) *stmt*

param → *type id*

param.kind = *value*

param → **ref** *type id*

param.kind = *ref*

param → **result** *type id*

param.kind = *result*

call → **id** (*exp*)

call.ok =

if (*lookupkind(id.name)* = *ref* **or**

lookupkind(id.name) = *result*)

then (*exp.kind* = *var*)

else *true*

*exp*₁ → **id** [*exp*₂]

*exp*₁.*kind* = *var*

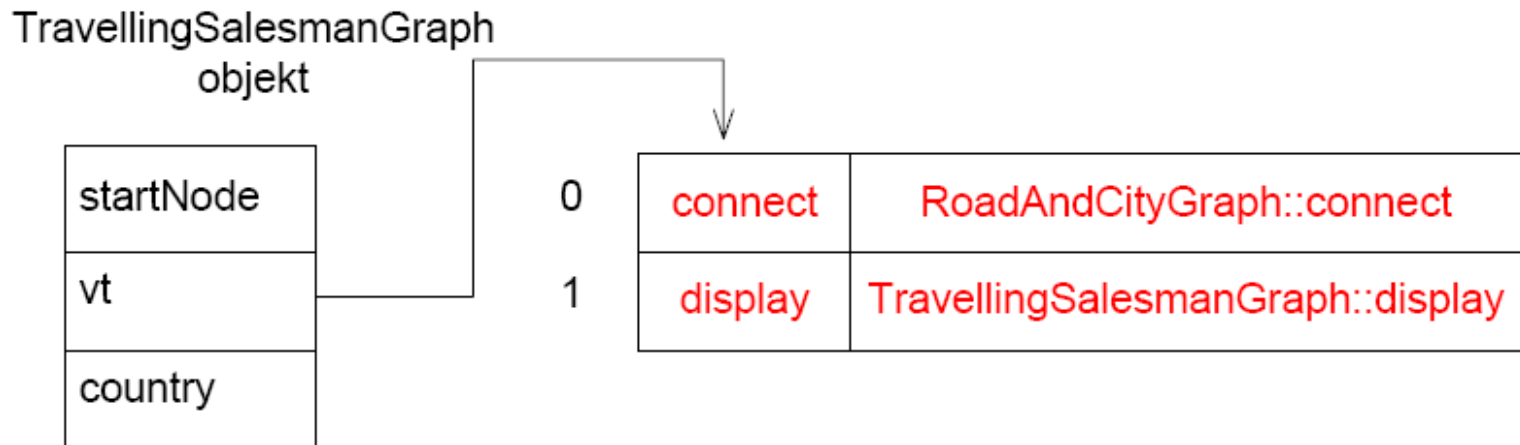
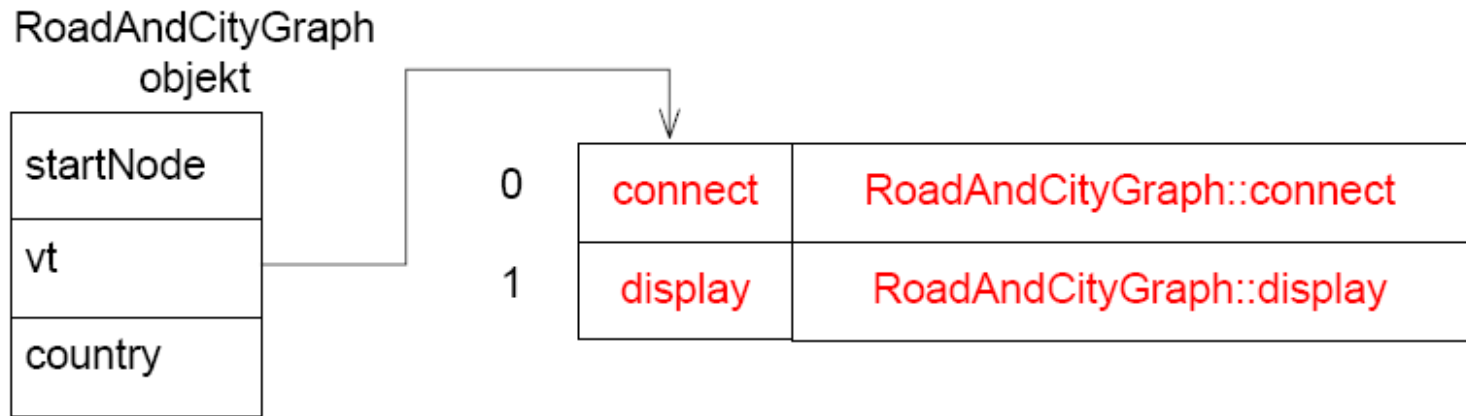
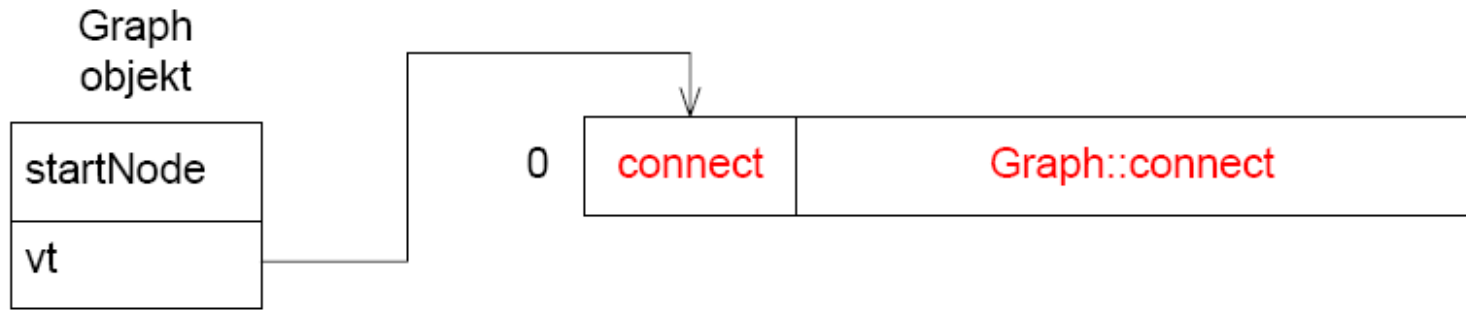
exp → **id**

exp.kind = *var*

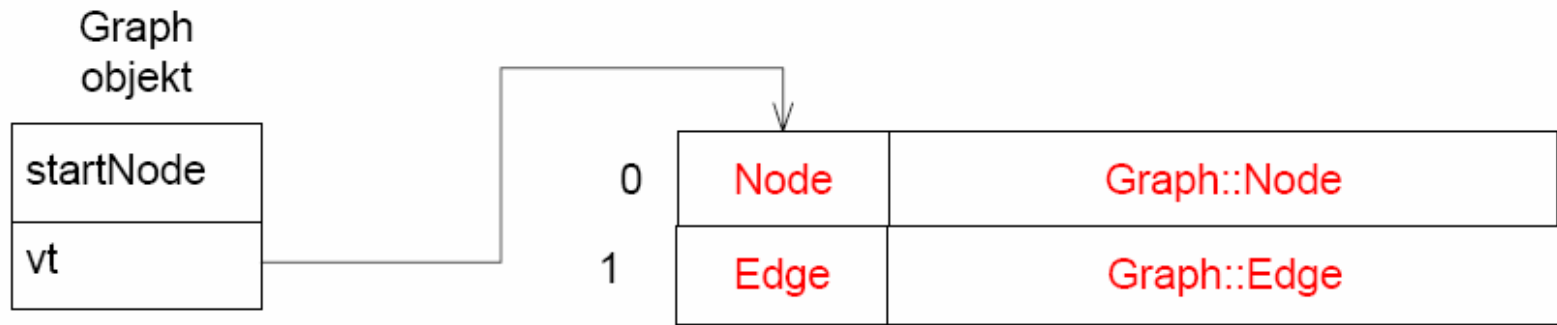
*exp*₁ → *exp*₂ *aritop* *exp*₃

*exp*₁.*kind* = *value*

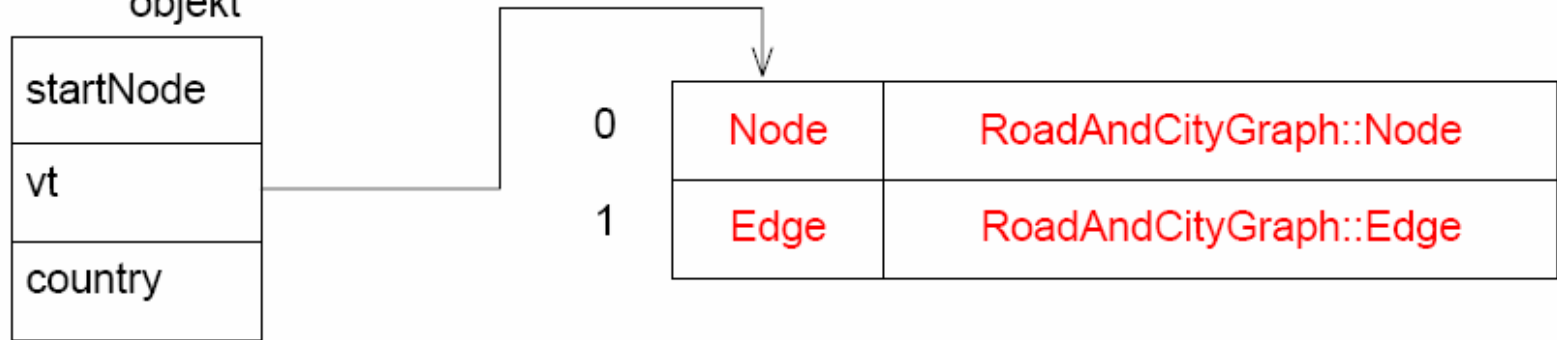
3a)



■ 3b)



RoadAndCityGraph objekt



TravellingSalesmanGraph objekt

