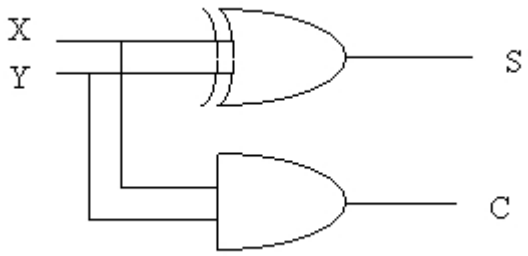
# Ασκήσεις Φυλλάδιο 1

## 1. Ψηφιακά Κυκλώματα

Οι πύλες είναι βασικά ηλεκτρονικά κυκλώματα με ένα αριθμό εισόδων και εξόδων, όπου η έξοδος ορίζεται από τις τιμές εισόδου. Στην Prolog ο πίνακας αληθείας μιας πύλης AND μπορεί να αναπαρασταθεί ως ακολούθως:

| x | y | x and y | Prolog |
|---|---|---|---|
| 0 | 0 | 0 | *and(0,0,0).* |
| 0 | 1 | 0 | *and(0,1,0).* |
| 1 | 0 | 0 | *and(1,0,0).* |
| 1 | 1 | 1 | *and(1,1,1).* |

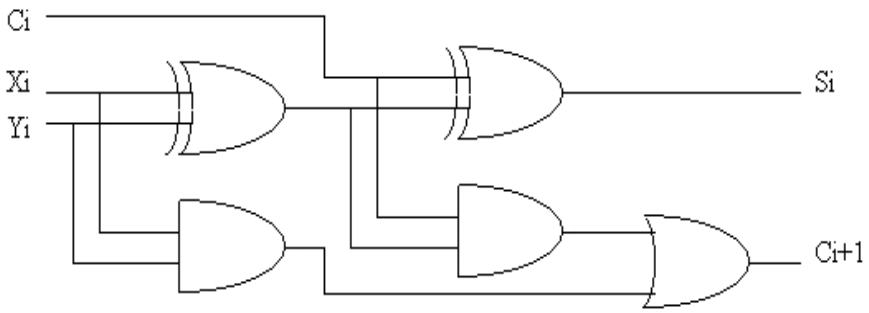
**1.1** Ένα ψηφιακό κύκλωμα είναι ένα σύνολο από συνδεδεμένες πύλες. Για παράδειγμα το παρακάτω κύκλωμα είναι ένας ημιαθροιστής (half adder) που δημιουργείται από AND και XOR πύλες.



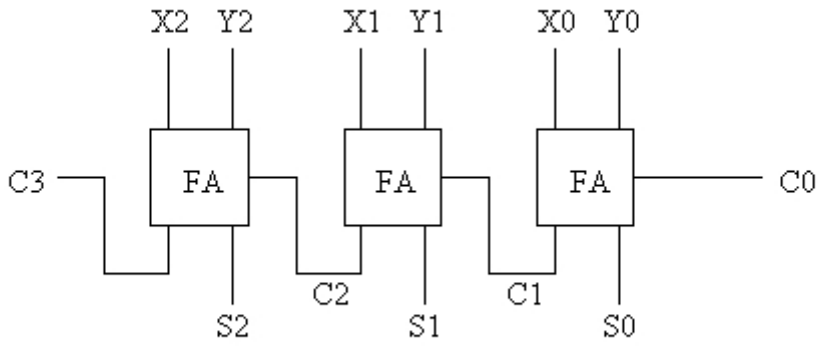

**a)** Γράψτε τον Prolog κώδικα που αντιστοιχεί στον παραπάνω ημιαθροιστή.
(**TIP**: *Επειδή η Eclipse Prolog έχει ήδη ένα κατηγόρημα xor/3, η πύλη xor να κωδικοποιηθεί σε ένα κατηγόρημα xor_gate/3.)*
**b)** Δώστε μια ερώτηση της Prolog που να αντιστοιχεί στις παρακάτω ερωτήσεις:

- Ποιος είναι ο πίνακας αληθείας του ημι-αθροιστή?
- Ποιες τιμές δίνουν κρατούμενο (C - Carry).

**1.2 a)** Δώστε ένα Prolog πρόγραμμα που να υλοποιεί τον ακόλουθο αθροιστή (full-adder):



**1.3 a)** Αναπτύξτε τον κώδικα Prolog για τον ακόλουθο 3-bit αθροιστή:




**b)** Δοκιμάστε τον κώδικά σας με είσοδο του αριθμούς X = 101 ( X0=1, X1=0, X2=1) και Y = 001 (Y0=1, Y1=0, Y2=0 and C0=0). Το αποτέλεσμα θα πρέπει να είναι S=110, i.e. S0=0, S1=1, S2=1 και C3=0.

## 2. The Company

Σας δίνεται το αρχείο **company.ecl** το οποίο περιέχει πληροφορίες για τους υπαλλήλους μιας εταιρείας, ως γεγονότα. Τα ακόλουθα είναι μερικά παραδείγματα για να γίνει κατανοητή η πληροφορία που υπάρχει στα αντίστοιχα γεγονότα του αρχείου.
Στοιχεία εργαζόμενου:

```
employee(name(john),position(programmer),wage(40000)).
```

- Δηλώνεται ότι ο υπάλληλος john, εργάζεται ως προγρραμματιστής (programmer) με ετήσιο μισθό 40000 ευρώ.

```
employee(name(alice),occupation(programmer),wage(35000)).
```

- Δηλώνει ότι η υπάλληλος alice, εργάζεται ως προγραμματίστρια με ετήσιο μισθό 35000 ευρώ.

Προσωπικά στοιχεία εργαζόμενου:

```
data(john,status(married,children(2))).
```

- Δηλώνει ότι ο john είναι παντρεμένος και έχει 2 παιδιά.

```
data(alice,status(single,children(0))).
```

- Δηλώνει ότι η alice δεν είναι παντρεμένη και δεν έχει παιδία.

α) Να υλοποιήσετε ένα κατηγόρημα **wage/2 (wage(Empl,Wage)**, το οποίο πετυχαίνει όταν ο Empl είναι υπάλληλος τη εταιρείας με ετήσιο μισθό Wage. Κατά την οπισθοδρόμηση θα επιστρέφονται όλες οι λύσεις. Για παράδειγμα:

```
?- wage(Empl, Wage).
Empl = john
Wage = 40000
Yes (0.00s cpu, solution 1, maybe more)

Empl = alice
Wage = 35000
```

```
Yes (0.00s cpu, solution 2, maybe more)

Empl = peter
Wage = 25000
Yes (0.01s cpu, solution 3, maybe more)
... (υπάρχουν και άλλες λύσεις)
```

β) Να υλοποιήσετε ένα κατηγόρημα **single_with_children(Empl,N)**, το οποίο πετυχαίνει αν ο υπάλληλος Empl δεν είναι παντρεμένος(single) και έχει N παιδιά, όπου N > 0. Για παράδειγμα:

```
?- single_with_children(Empl, N).
Empl = helen
N = 2
Yes (0.00s cpu, solution 1, maybe more)
Empl = donald
N = 1
Yes (0.00s cpu, solution 2, maybe more)
No (0.00s cpu)
```

γ) Να υλοποιήσετε ένα κατηγόρημα **benefit(Name,Wage,Benefit)** το οποίο πετυχαίνει όταν ο υπάλληλος Name, με ετήσιο μισθό Wage, παίρνει επίδομα Benefit. Το επίδομα καθορίζεται ως εξής:

- Αν ο υπάλληλος δεν είναι παντρεμένος και δεν έχει παιδιά το επίδομα είναι 0.
- Αν ο υπάλληλος δεν είναι παντρεμένος και έχει παιδιά, το επίδομα είναι 1500.
- Αν ο υπάλληλος είναι παντρεμένος χωρίς παιδιά το επίδομα είναι 500.
- Αν ο υπάλληλος είναι παντρεμένος και έχει παιδιά το επίδομα είναι 1100 ευρώ.

Για παράδειγμα:

```
?- benefit(john, W, B).
W = 40000
B = 1100
Yes (0.00s cpu, solution 1, maybe more)

?- benefit(helen, W, B).
W = 140000
B = 1500
Yes (0.00s cpu, solution 1, maybe more)

?- benefit(alice, W, B).
W = 35000
B = 0
Yes (0.00s cpu, solution 1, maybe more)
No (0.29s cpu)
```

## 3. Το πρόβλημα των 4 επαγγελμάτων (Smith, Baker, Carpenter and Tailor Problem)

**1.** Smith, Baker, Carpenter and Tailor have each a different profession (smith, baker, carpenter, tailor) but not showed by their names, i.e. the Smith is not a smith, etc. Each of them has a son. But the sons do not have the profession showed by their name, either.

If you know that:

1) no son has the same profession as his father has

2) the Baker has the same profession as the Carpenter's son has

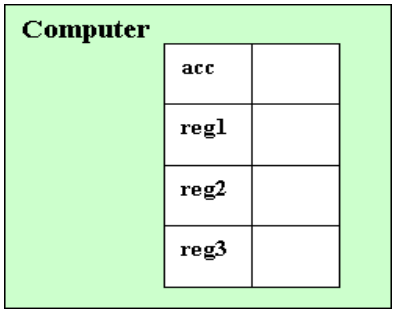
3) Smith's son is a baker.

find the professions of the parents and sons. Write a prolog predicate *professions/8* that finds the professions of the different people mentioned above.

**Σημείωση: Θα χρειαστείτε το κατηγόρημα \= (not unify) το οποίο πετυχαίνει όταν οι δύο όροι δεν μπορούν να ενοποιηθούν.**

## 4. Αυτόματη Παραγωγή Κώδικα

Έστω μια μηχανή με ένα ειδικό συσσωρευτή (accumulator - acc) και τρεις γενικής χρήσης καταχωρητές (reg1, reg2, reg3), όπως φαίνεται στο ακόλουθο σχήμα:




Η κατάσταση μιας τέτοιας μηχανής μπορεί να αναπαρασταθεί από τον σύνθετο όρο:

***state(acc(<περιεχ. acc>),reg1(<περιεχ. reg1>),reg2(<περιεχ. reg2>),reg3(<περιεχ. reg3>))***

Οι διαθέσιμες εντολές στη μηχανή είναι οι ακόλουθες:

- **add R:** πρόσθεση του περιεχομένου του καταχωρητή R στον συσσωρευτή.
- **subtract R:** αφαίρεση του περιεχομένου του καταχωρητή R από τον συσσωρευτή.
- **store R:** αποθήκευση του περιεχομένου του συσσωρευτή στον καταχωρητή R.
- **load R:** αποθήκευση του περιεχομένου του καταχωρητή R στον συσσωρευτή.

Οι παραπάνω εντολές μπορούν εύκολα να αναπαρασταθούν στην Prolog. Για παράδειγμα το ακόλουθο γεγονός αναπαριστά την πράξη της πρόσθεσης ανάμεσα του περιεχομένου του καταχωρητή reg1 και του συσσωρευτή με αποθήκευση του αποτελέσματος στον συσσωρευτή.

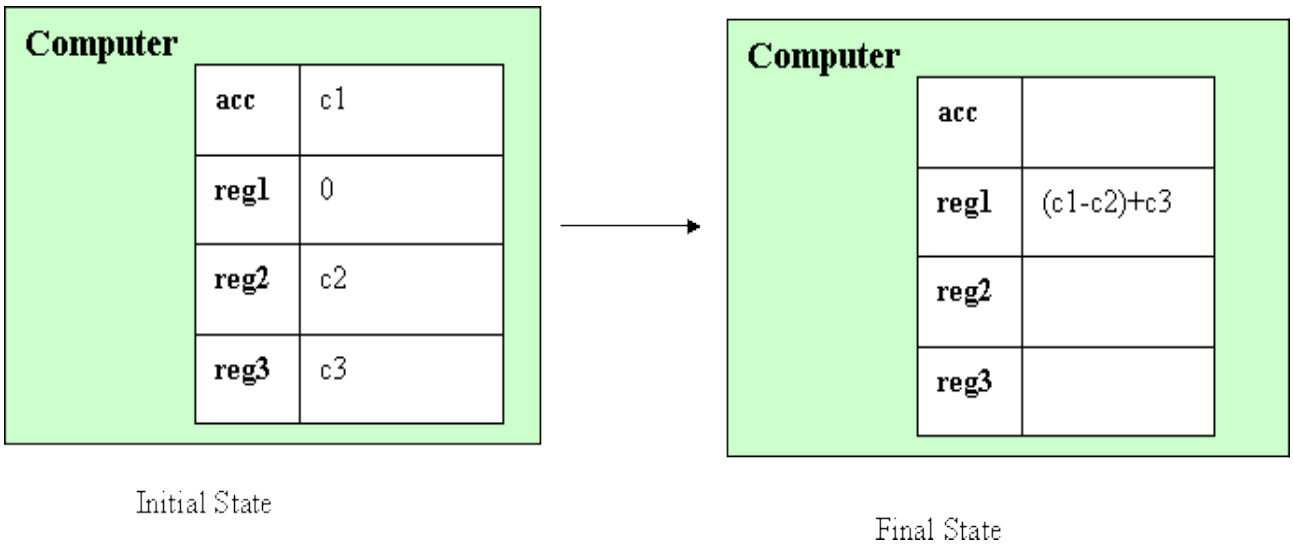
***command(add_r1,state(acc(X),reg1(Y),R2,R3),state(acc(X+Y),reg1(Y),R2,R3)).***

**1.** Κωδικοποιήστε όλες τις παραπάνω πράξεις σαν Prolog γεγονότα. Προσοχή, θα χρειαστείτε μια πράξη ανά καταχωρητή (δηλ add_r1, add_r2, add_r3 κλπ).

**2.** Αν η αρχική και η τελική κατάσταση της μηχανής φαίνονται στο ακόλουθο σχήμα, και είναι γνωστό ότι απαιτούνται μόνο τρεις εντολές μηχανής για να μεταβούμε από την πρώτη στη δεύτερη, ορίστε ένα κατηγόρημα **findOps/3,** το οποίο επιστρέφει στα 3 ορίσματά του τις εντολές που απαιτούνται.

Παράδειγμα εκτέλεσης:

```
?- findOps(O1, O2, O3).
O1 = subtract_r2
O2 = add_r3
O3 = store_r1
```

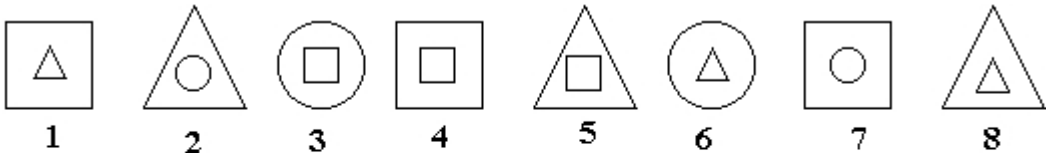



Initial State

Final State

## 5. Το πρόβλημα των Αναλογιών (IQ Test)

Στα κλασικά IQ τεστ, εμφανίζεται συχνά το πρόβλημα των γεωμετρικών αναλογιών. Έστω τα σχήματα που ακολουθούν με την αρίθμηση που φαίνεται παρακάτω. Η κλασσική ερώτηση που υπάρχει σε τέτοιου είδους τεστ είναι η ακόλουθη: ***"Αν το σχήμα Α σχετίζεται με το σχήμα Β, τότε ποιο σχήμα σχετίζεται με το σχήμα Γ με την ίδια σχέση;"***




Στη εικόνα παραπάνω αν τα σχήματα 1 και 5 σχετίζονται, και δοθεί το σχήμα 3, τότε προφανώς η απάντηση είναι στο σχήμα 7. Το σχήμα 1 είναι ένα τρίγωνο μέσα σε ένα τετράγωνο, το σχήμα 5 ένα τετράγωνο μέσα σε ένα τρίγωνο, άρα υπακούν στη σχέση inverse (ανάστροφο). Εφόσον το σχήμα 3 είναι ένα τετράγωνο μέσα σε ένα κύκλο, το ανάστροφό του είναι το 7 (ένας κύκλος σε ένα τετυράγωνο).

Η αναπαράσταση των σχημάτων μπορεί να γίνει με γεγονότα της μορφής:

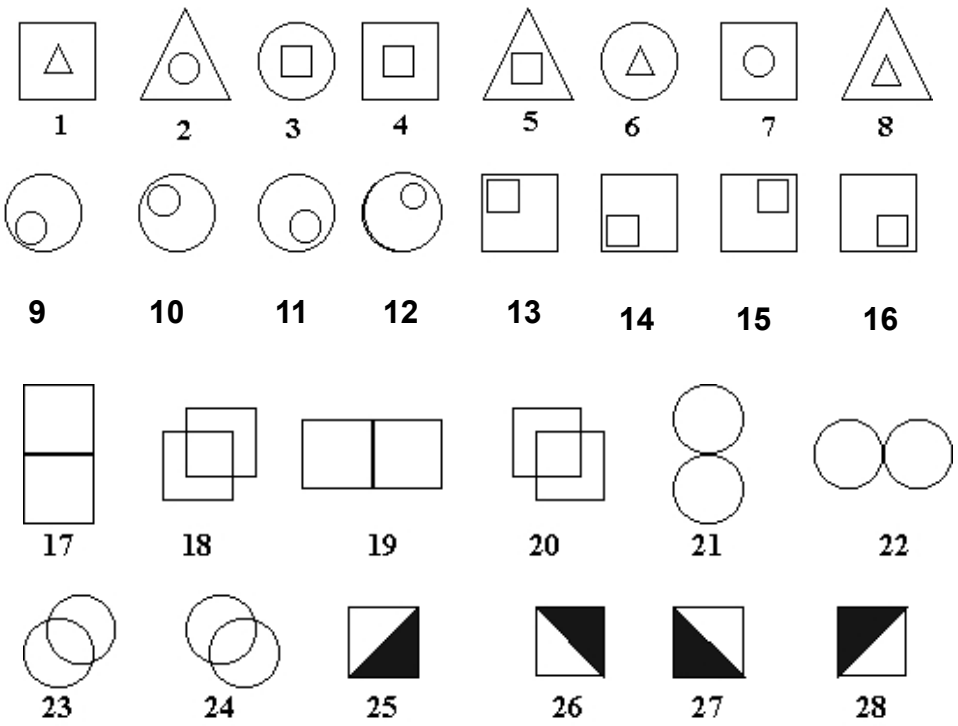
***figure(1, middle(triangle, square)).***

Οι σχέσεις μπορούν να κωδικοποιηθούν σαν

***relation( middle(S1,S2), middle(S2,S1), inverse).***

η οποία λέει απλά ότι δύο σχήματα ακολουθούν τη σχέση inverse αν το ένα αποτελείται από το S1 στο κέντρο του S2, και το άλλο από το S2 στο κέντρο του S1.

**1**. Γράψτε όλα τα κατηγορήματα (γεγονότα) *figure/2* για τα **πρώτα 16 σχήματα** που φαίνονται παρακάτω. (τα υπόλοιπα αν θέλετε να "δοκιμαστείτε" παραπάνω).




**2.** Γράψτε όλες τις πιθανές σχέσεις σαν γεγονότα *relation/3* που μπορείτε να σκεφτείτε.

**3.** Γράψτε ένα κατηγόρημα *analogy/4*, που αν κληθεί επιλύει την ερώτηση των γεωμετρικών αναλογιών. Για παράδειγμα:

***?- analogy(1,5,3,X).***
***X=7***

**Hint**: Πρέπει να βρείτε πρώτα την σχέση που διέπει τα δύο σχήματα που αντιστοιχούν στους αριθμούς των δύο πρώτων ορισμάτων και μετά να "υπολογίσετε" ποιο είναι το κατάλληλο σχήμα με βάση τη σχέση που βρήκατε και το τρίτο όρισμα και τέλος να υπολογίσετε τον αριθμό του σχήματος που αποτελεί την απάντηση.