Prolog

Operators:

=	Unification of right and left terms
==	Syntactical identity
=:=	Arithmetic equality
is	Arithmetic instanciation
\== ou \=	Syntactical difference
=\=	Arithmetic difference
+	Addition
-	Substraction
*	Multiplication
/	Real division
//	Integer division
mod	Remaining of Euclidean division
< > =< =>	Arithmetic comparison

Prolog

Operators

- Unification :

```
?- X+1=2+1.
X = 2.

?- 2+X=3+1.
false.
```

- Syntactical identity:

```
?- X==X.
true.

?- dad==mam.
false.

?- "dad"=="dad".
true.

?- dad==dad.
true.
```

- Arithmetic equality:

```
?- 1+3=:=4.
true.

?- 1+3=:=3+1.
true.
```

- Arithmetic instanciation :

```
?- X is 3+4.
X = 7.

?- X is +(3,4).
X = 7.

?- X is 3*4+4*5.
X = 32.

?- X is +(*(3,4),*(4,5)).
X = 32.
```

- Syntactical difference:

```
?- X\=Y.   (variables !)
false.

?- dad\=mam.
true.

?- dad\==mam.
true.

?- dad\=dad.
false.
```

- Arithmetic difference:

```
?- 3=\=6.
true.

?- 3=\=3.
false.
```

Prolog

Structure of Prolog programme :

- Facs
- Rules
- Queries (Goals)

Utilized symbols:

Predicate (or function, constant)	Word or letter in lowercase
Variable	Word in uppercase (or starting by capital letter or by _)
:-	**if** Exp. individual(X):-person(X). read: X is an individual if X is a person
,	**and**
;	**Or**
not	**Negation**
_	**Anonymous variable(∀)**

Prolog

Predicates to Prolog

- All dogs are animals
- All animals will die
- Fido is a dog

Deductions:

- All dogs will die
- Fido will die
- Fido is an animal

Prolog code (see file: chien.pl)

Prolog

Predicates to Prolog

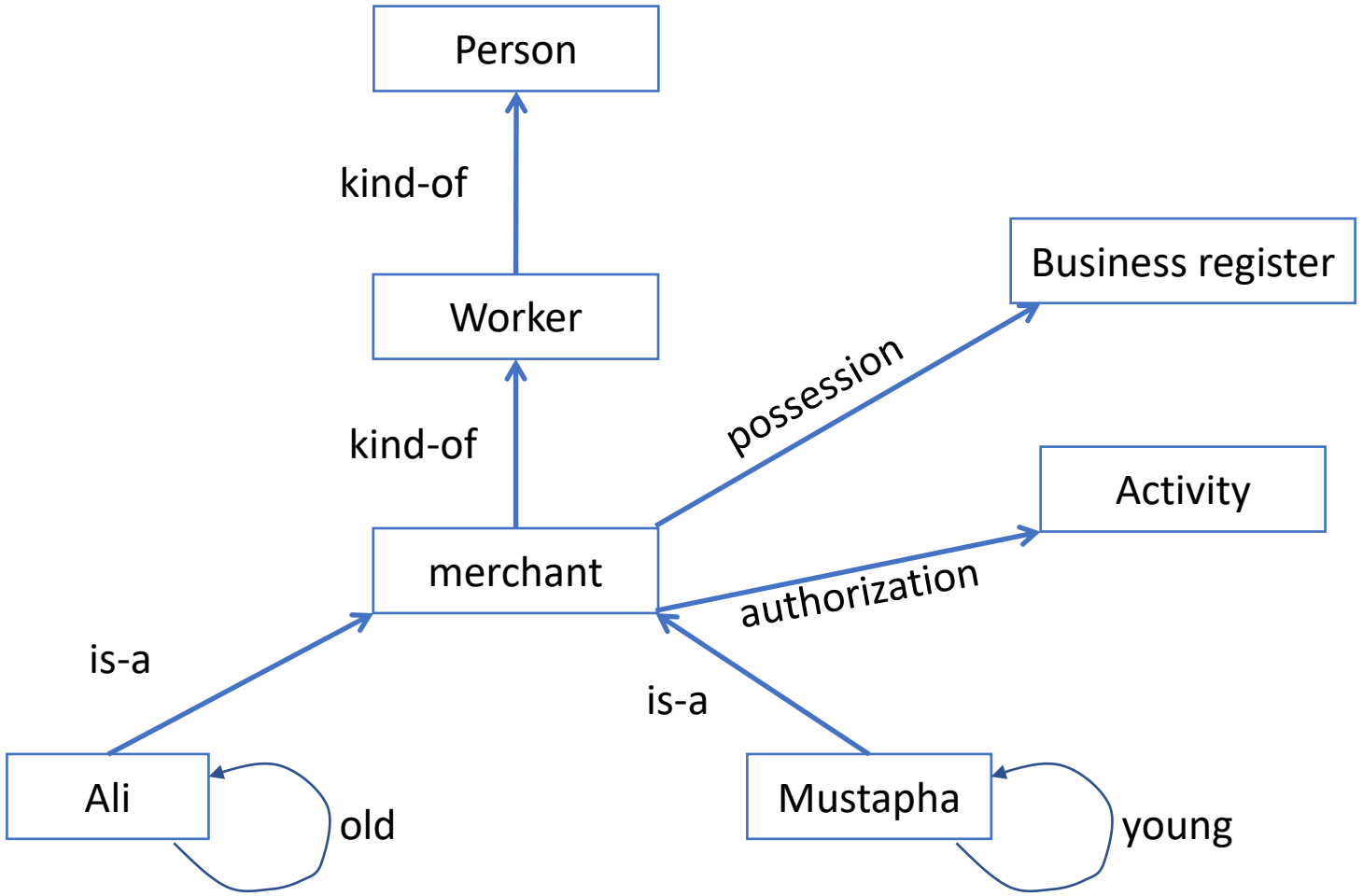
Individual / Person :

- **Create some persons (define age of each of them).**
- **An individual is a person whatever is his age.**
- **A minor is a person having less than 18 years old.**

Prolog code(see file : personne.pl)

Prolog

Semantic net to Prolog

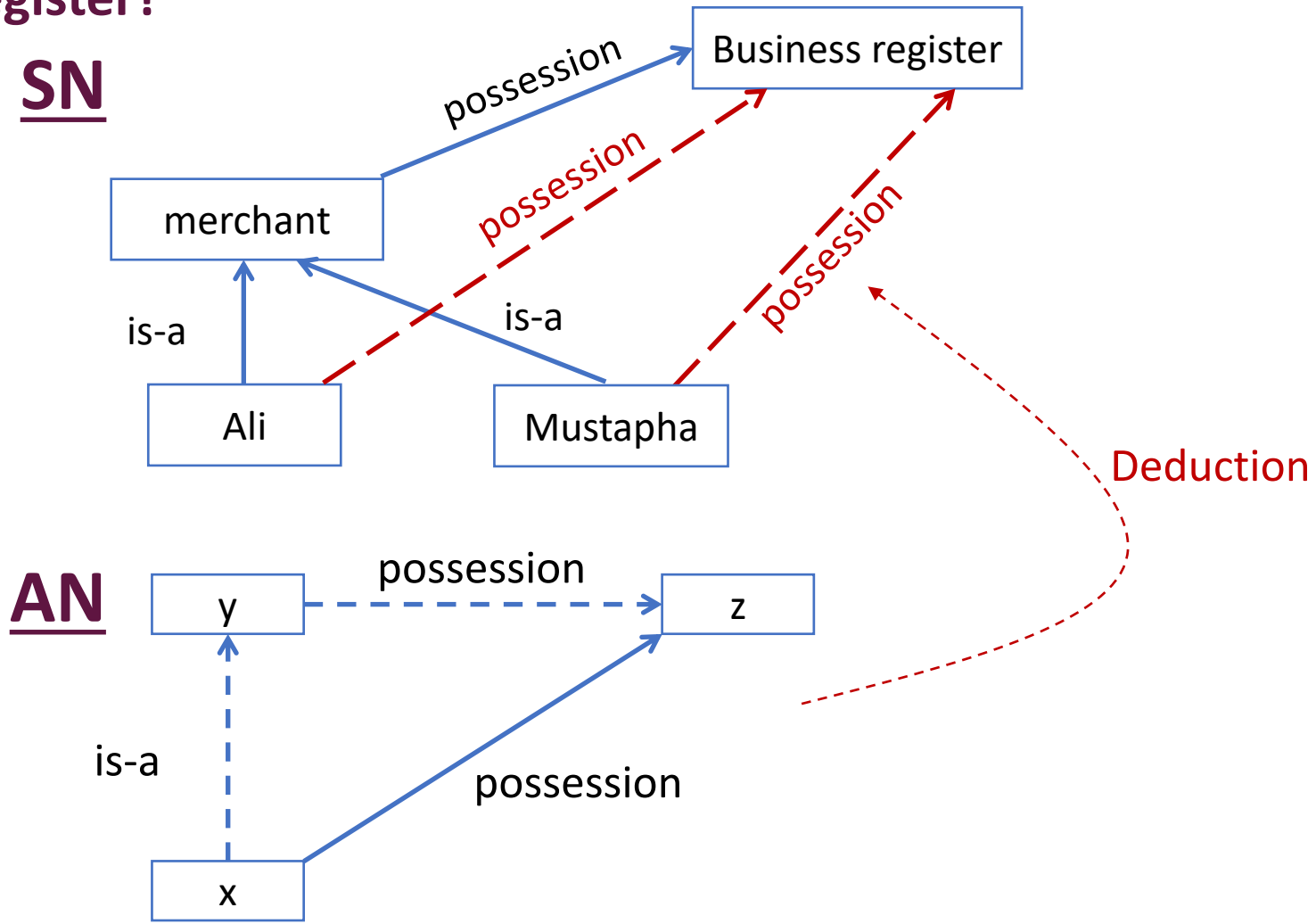


Prolog code (see file: commerce.pl)

Prolog

Use of auxilliary Semantic net

Who has a business register?



Prolog code (see file: commerce.pl)