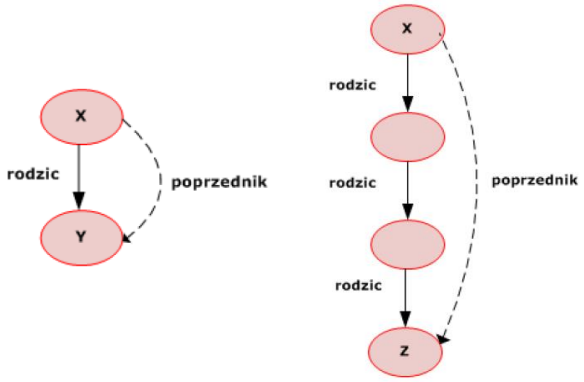
Reguły rekurencyjne

Procedura rekurencyjna to zespół kilku klauzul opartych o ten sam predykat. Dla procedur rekurencyjnych są to dwie reguły:

- Fakt powodujący zakończenie rekurencji.
- Regułę, w której nagłówku i ciele znajduje się ten sam predykat, ale z innymi argumentami.

Bazując na programie opisującym rodzinę wprowadźmy jeszcze jedną relację poprzednik. Definicja tej relacji składać się będzie z dwóch reguł: pierwszej opisującej bezpośredniego poprzednika i drugiej opisującej dalsze pokrewieństwo. Przykłady obu typów relacji poprzednik pokazano na rysunku poniżej.



Pierwszą regułę zapisujemy w bardzo prosty sposób:

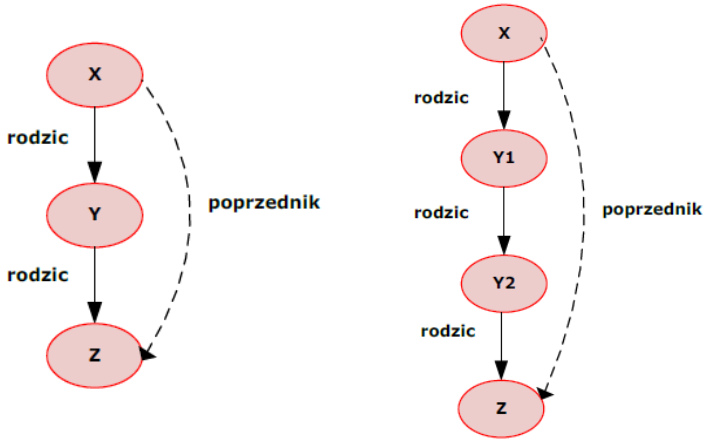
```
Dla każdych X i Y
      X jest poprzednikiem Y,
      jeśli X jest rodzicem Y.
```

Co w Prologu przybiera postać:

```
poprzednik(X, Y) :- rodzic(X, Y).
```

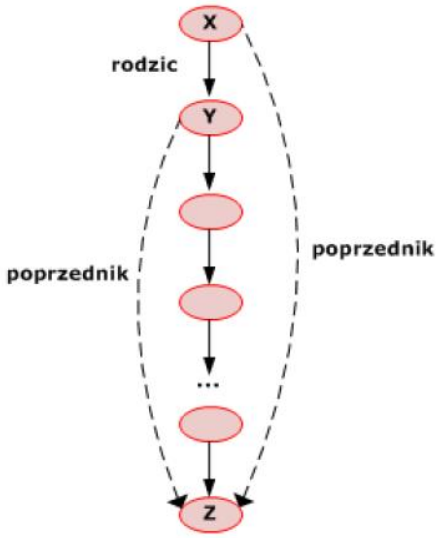
Nieco trudności może pojawić się przy zapisaniu drugiej reguły tzn. bycia nie rodzicem, lecz dziadkami, pradziadkami itp. Zauważmy, że poprzednicy mogą być z dowolnego pokolenia wstecz: 2, 3, 4, …, n. Nasza reguła powinna być na tyle uniwersalna, żeby obejmować wszystkie te przypadki. Reguła zapisana zgodnie z rysunkiem obejmuje tylko przypadek bycia poprzednikiem „3- go stopnia".

```
poprzednik(X, Z) :- rodzic(X, Y1), rodzic(Y1, Y2), rodzic(Y2, Z).
```



Na razie mamy zdefiniowane relacje do 3-go stopnia włącznie. Widać, więc, że trzeba spróbować do problemu podejść w nieco inny sposób. Zaproponowana przez nas metoda nigdy nie będzie uniwersalna. Zawsze możemy zapytać o poprzednika takiego stopnia, że nie będziemy mieli

odpowiedniej reguły stworzonej w programie. Spróbujmy wyrazić relację bycia przodkiem przy użyciu relacji przodek, bo czy nie prawdziwe jest poniższe logiczne stwierdzenie:




```
Dla każdych X i Z,
  X jest poprzednikiem Z, jeśli
        istnieje takie Y, że
        1) X jest rodzicem Y i
        2) Y jest poprzednikiem Z.
```

Co zapisane w sposób sformalizowany przybiera postać:

```
  poprzednik(X, Z):-
     rodzic(X, Y),
     poprzednik(Y, Z).
```

Posiadamy kompletną definicję bycia przodkiem składającą się z dwóch reguł:

```
poprzednik(X, Y) :-
      rodzic(X, Y).

poprzednik(X, Z) :-
      rodzic(X, Y),
      poprzednik(Y, Z).
```

Reguły takie jak przedstawiona powyżej nazywamy ***rekurencyjnymi.*** Mechanizm rekurencji to jeden z najważniejszych mechanizmów występujących w Prologu. Nie ma możliwości napisania żadnego bardziej skomplikowanego programu bez użycia rekurencji. Poprawność stworzonej reguły rekurencyjnej możemy łatwo sprawdzić zadając odpowiednie pytanie:
Kim byli przodkowie Mikołaja?

```
? - poprzednik(X, mikołaj).
```

Uzyskujemy poprawną odpowiedź:

```
X = krzyś;
X = zosia;
X = andrzej;
X = marcin;
```

Zastanówmy się, w jaki sposób są znajdowane rozwiązania. W tym celu złóżmy nasz program w jedną całość, dodając komentarze pozwalające zrozumieć nasz program.

Komentarz to ciąg znaków dowolnej długości zawarty pomiędzy symbolami `/* i */` lub po znaku `%` np.:

- `/* Komentarz może być dowolnej długości i znajdować się w kilku liniach programu. Spróbuj dodawać komentarze, żeby ułatwić innym korzystanie z programu */`
- `% tak też może wyglądać komentarz`

```
rodzic(zosia, marcin).          /*Zosia jest rodzicem Marcina */
rodzic(andrzej, marcin).
rodzic(andrzej, kasia).
rodzic(marcin, ania).
rodzic(marcin, krzyś).
rodzic(krzyś, mikołaj).
mężczyzna(andrzej).        /* Andrzej jest mężczyzną*/
mężczyzna(marcin).
mężczyzna(krzyś).
mężczyzna(mikołaj).
kobieta(kasia).            /* Kasia jest kobietą*/
kobieta(ania).
kobieta(zosia).
potomek(Y, X):-            /* Y jest potomkiem X*/
     rodzic(X, Y).         /* X jest rodzicem Y*/
matka(X, Y):-              /* …*/
     rodzic(X, Y),
     kobieta(X).
dziadkowie(X, Z):-         /* ….*/
     rodzic(X, Y),
     rodzic(Y, Z).
siostra(X, Y):-            /* ….*/
     rodzic(Z, X),
     rodzic(Z, Y),
     kobieta(X),
     X \= Y.
poprzednik(X, Y): -        /* reguła r1*/
     rodzic(X, Y).
poprzednik(X, Z):-         /* reguła r2*/
     rodzic(X, Y),
     poprzednik(Y, Z).
```

Poszukiwanie rozwiązań

Pytanie w Prologu jest zawsze ciągiem jednego lub więcej celów. Odpowiadając na zadane pytanie Prolog stara się wykazać, że podany cel jest prawdziwy, zakładając, że wszystkie podane w bazie wiedzy fakty i reguły są prawdziwe. Inaczej mówiąc, wykazanie prawdziwości celu jest równoważne z logicznym wywnioskowaniem celu na podstawie podanych faktów i reguł. Jeśli dodatkowo pytanie zawiera zmienne Prolog stara się je ukonkretnić do poszczególnych obiektów, które następnie są wyświetlane użytkownikowi. Inna jeszcze interpretacja programu w Prologu mówi, że fakty i reguły to zbiór aksjomatów, natomiast cel jest pewną hipotezą, którą należy wykazać.

Prześledźmy teraz sposób wnioskowania programu na podstawie programu opisującego relacje rodzinne:

```
?- poprzednik(andrzej, krzyś).
```

Wiemy, że `rodzic(marcin, krzyś)` zachodzi, ponieważ jest to fakt z naszej bazy. Za pomocą tego faktu i reguły `r1` można wyciągnąć wniosek, że zachodzi `poprzednik(marcin, krzyś).` Nie mamy wprost zapisanego takiego faktu w bazie wiedzy, ale możemy go wywnioskować. W bardziej eleganckiej formie możemy zapisać, że:

```
rodzic(marcin, krzyś) ==> poprzednik(marcin, krzyś)
```

Kolejno wiemy że `rodzic(andrzej, marcin)` to fakt. Za pomocą tego faktu i wyprowadzonego wcześniej stwierdzenia `poprzednik(marcin, krzyś)` możemy stwierdzić, że zachodzi `poprzednik(andrzej, krzyś)` na postawie reguły `r2`. Możemy cały tok rozumowania zapisać teraz następująco:

```
rodzic(marcin, krzyś) ==> poprzednik(marcin, krzyś)
rodzic(andrzej, marcin) i poprzednik(marcin, krzyś) ==>
poprzednik(andrzej, krzyś)
```

W taki sposób pokazano logiczny ciąg kroków, do wykazania celu. Nie wiemy dalej jednak jak Prolog znajduje dokładnie ten ciąg kroków. Formalnie, aby wykazać w języku klauzul, że zbiór założeń implikuje pewną konkluzję, zakłada się, że konkluzja nie jest prawdziwa i pokazuje, że negacja konkluzji jest sprzeczna z założeniami, co oczywiście dowodzi prawdziwości konkluzji.

Zastanówmy się jak będzie wyglądał proces dowodzenia, jeśli pytanie brzmi:

```
? - poprzednik(andrzej, krzyś).
```

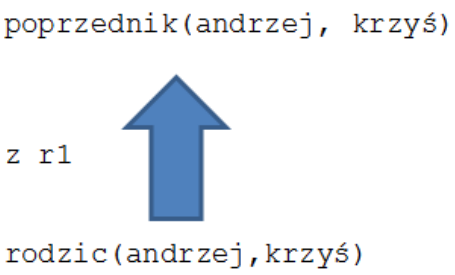
W celu wykazania poprawności powyższego stwierdzenia Prolog stara się znaleźć klauzulę, z której powyższy cel można wyprowadzić. Jedyne klauzule odnoszące się do pojęcia przodek to reguły oznaczone przez nas jako `r1` i `r2`. Mówimy, że uzgadniamy cel z głową reguły. Ponieważ mamy dwie klauzule odnoszące się do relacji przodek, pytanie którą wybierze Prolog. Najpierw zostanie zastosowana reguła, która jako pierwsza występuje w programie (r1).

```
poprzednik(X, Z) :- rodzic(X, Z).
```

Ponieważ chcemy wykazać prawdziwość celu `poprzednik(andrzej, krzyś)` zmienne w regule przyjmą konkretną wartość (zostaną ukonkretnione) w następujący sposób:

```
X = andrzej, Y = krzyś
```

Oryginalny cel `poprzednik(andrzej, krzyś)` zostaje zastąpiony przez podcel `rodzic(andrzej,krzyś)`, co przedstawiono na rys.



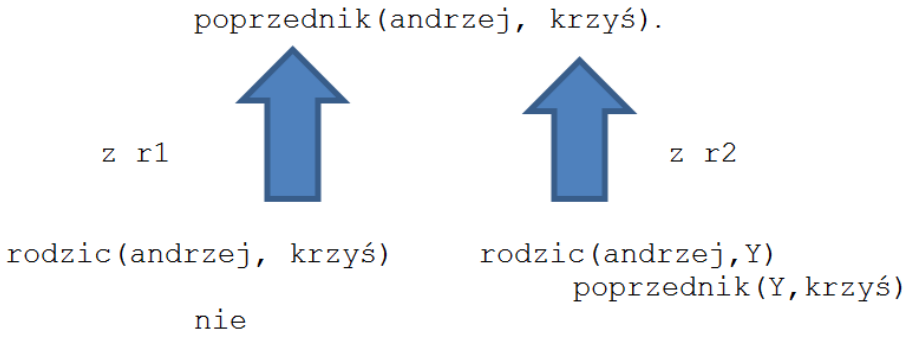
Ponieważ nie ma w programie klauzuli, którą można **uzgodnić** (Uzgadnianie zmiennych to próba sprawienia, by były sobie równe) z celem `rodzic(andrzej, krzyś)` skorzystanie z reguły `r1` nie powiodło się. Teraz Prolog wykonuje operacje nawracania do pierwszego celu, żeby spróbować alternatywnego sposobu na dowiedzenie pytania `poprzednik(andrzej, krzyś)`. Reguła `r2` zostaje teraz zastosowana:

```
poprzednik(X, Z) :- rodzic(X, Y), poprzednik(Y, Z).
```

Jak poprzednio zmienne X i Z zostają ukonkretnione (ukonkretnienie to nadanie konkretnej wartości zmiennym) `X = andrzej, Z = krzyś.`

Zauważmy, że w powyższej regule zmienna Y nie została jeszcze ukonkretniona. Główny cel zostaje zamieniony na dwa cele (patrz poniższy rysunek).

```
rodzic(andrzej, Y), poprzednik(Y, krzyś)
```



Stojąc teraz wobec dwóch celów do wykazanie, Prolog będzie się starał je pokazać zgodnie z kolejnością ich występowania. Pierwszy z celów jest bardzo łatwy do wykazania. W wyniku uzgodnienia zmienna Y zostaje ukonkretniona.

```
Y = marcin
```

Pierwszy cel został dowiedziony, został jeszcze cel drugi, który przyjmuje postać:

```
poprzednik(marcin, krzyś)
```

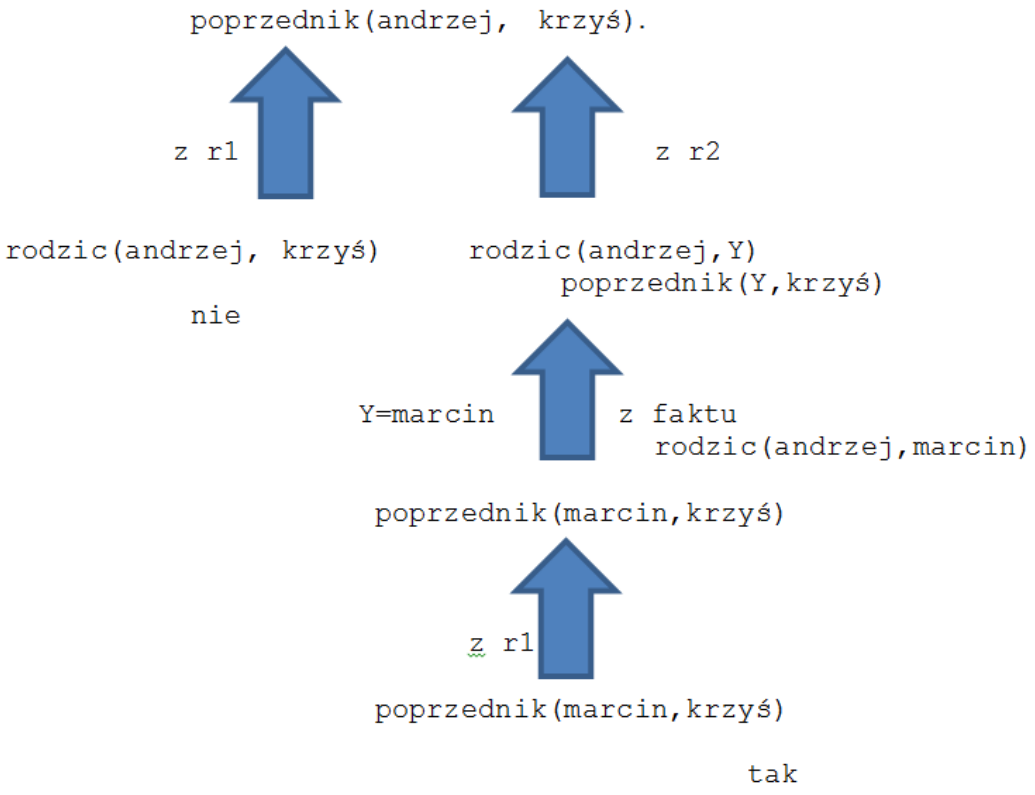
Żeby wykazać powyższy cel ponownie korzystamy z reguły r1. Jest to drugie użycie tej reguły i nie ma żadnego związku z poprzednim wywołaniem. Prolog za każdym razem wywołuje regułę z nowymi zmiennymi. Żeby podkreślić ten fakt przepiszmy regułę za pomocą nowych zmiennych:

```
poprzednik(X', Z') :- rodzic(X', Z').
```

Po ukonkretnieniu zmiennych otrzymujemy:

```
X' = marcin, Z' = krzyś
```

Cel poprzednik(marcin, krzyś) zostaje zamieniony na cel rodzic(marcin, krzyś). Ten cel jest spełniony, ponieważ występuje, jako prosty fakt w programie. Kompletny tok rozumowania został graficznie pokazany na rysunku poniżej.



Zadania do samodzielnego rozwiązania:

Zadanie 1.

Wykorzystując utworzoną uprzednio bazę wiedzy rodzina.pl rozwiąż poniższe zadania:

1. Zapisz w Prologu następujące stwierdzenia:
   - Każdy, kto ma dziecko jest szczęśliwy (wprowadź jednoargumentową relację szczęśliwy).
   - Dla każdego X, jeśli X ma dziecko, które ma siostrę wtedy X ma dwoje dzieci (wprowadź relację o nazwie dwoje_dzieci).
2. Zapisz relację wnuk za pomocą relacji rodzic.
3. Zdefiniuj regułę ciotka(X, Y) wykorzystując fakty rodzice i regułę siostra.
4. Zdefiniuj regułę następca.

Zadanie 2.

Stwórz bazę wiedzy zawierającą przykładowe osoby, z miejscem ich pracy, wielkością firmy (mała, średnia, duża) i wysokością zarobków, np.:

```
zatrudnienie(anna, nowak, zelmer, duza_firma, 2000).
```

Następnie utwórz regułę, która mówi, że dana osoba jest zadowolona z pracy, jeśli pracuje w małej firmie i zarabia więcej niż 2500 zł.

Sprawdź czy są w twojej bazie wiedzy osoby, które pracują w dużej firmie i zarabiają mniej niż 1500 zł.

Zadanie 3.

Dla danej bazy wiedzy samoloty.pl skonstruuj zapytania, które znajdą odpowiedź na poniższe pytania:
/*dla struktury lot podano kolejno informacje: nr lotu, miejsce początkowe, miejsce docelowe, godzina wylotu (1800 oznacza godzinę 18:00), godzina przylotu, dni w które kursuje samolot od poniedziałku do niedzieli (1 – jest samolot, 0 – brak połączenia)*/

1. Wypisać wszystkie przeloty, jakie mają miejsce we wtorek
2. Podać godziny przelotów na trasie rzeszow-warszawa
3. Sprawdzić, na jakiej trasie operuje lot o numerze a2324
4. Wypisać dni, w jakie kursuje samolot z berlina do warszawy
5. Jakie samoloty lądują na lotnisku w warszawie w godzinach od 8:00 do 10:00 lub po 18:00