Recursividad

Demostración por recursividad

- Generalización del concepto de inducción matemática: No es necesario que la inducción se haga sobre el conjunto de los números naturales.
- Se debe definir en cambio un ***dominio bien fundamentado***: Toda sucesión descendente debe ser finita.

Definiciones recursivas

- Se busca poder definir dominios bien fundamentados.
- La definición se puede aplicar repetidamente (recursivamente) hasta llegar al caso base.

Fuente: https://thomaspark.co/2017/01/relevant-xkcd/

Ejemplo 1

Definición “Descendiente de”

- Todos los hijos de x son descendientes de x.
- Si y es hijo de x, entonces los descendientes de y son descendientes de x.
- No hay nadie más que sea descendiente de x.

Ejemplo 2

Definición: Expresión lógica simplificada (SL)

- Todas las constantes y variables proposicionales son SL.
- Si A y B son SL, entonces $(A \wedge B)$, $(A \vee B)$ y $\neg A$ también son SL.
- No hay nada más que sea SL.

Ejemplo 3

Definición de expresión matemática simplificada (SM)

- Todos los enteros y nombres de variables son expresiones SM.
- Si A y B son expresiones SM, también lo son: -A, (A+B), (A-B), (AxB), (A/B).
- No hay nada más que sea una expresión SM.

Para explorar más allá:
Gramáticas para los lenguajes de programación
BNF and EBNF: What are they and how do they work?

Reglas de Alargamiento y Acortamiento

En una definición recursiva:

- Reglas que extienden un caso previo, en el sentido que el caso previo pasa a ser un subcomponente del nuevo caso, se llaman reglas de ***alargamiento***. E.g. SL, SM.
- Reglas que producen elementos que no son subcasos del caso nuevo, se llaman reglas de ***acortamiento***. E.g. Simplificaciones en aritmética, simplificaciones lógicas.

Definición recursiva de lista

- La lista vacía es una lista y se denota [].
- Si a es un nodo y B es una lista, entonces [a|B] es una lista. a es *la cabeza* y B es *la cola* de la lista.
- No hay nada más que sea una lista.

Ejemplos Prolog: Operaciones con listas

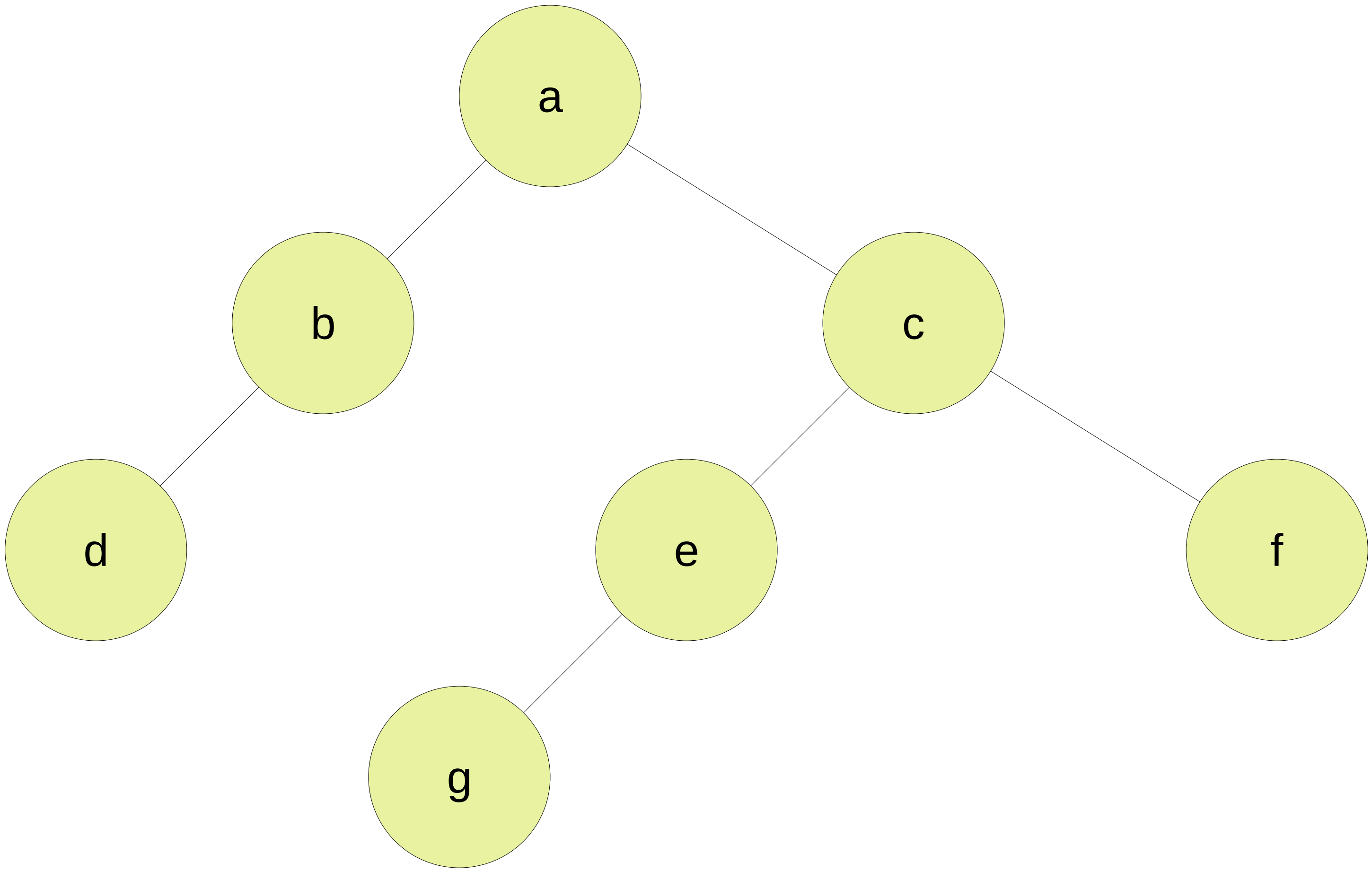
```
% Concatenación con un elemento
L=[a|[b|[]]].
% Notación simplificada (azúcar sintáctico)
L=[a|[b|[]]], L=[a,b].
% Unificación
[a|X]=[Y|[b]].
[a|X]=[a].
[Y|X]=[a,b,c].
```

Árboles

- Un árbol es una estructura que representa ciertas relaciones entre individuos: e.g. Árbol genealógico, organigrama de una empresa.
- Un árbol se compone de arcos/aristas y nodos.
- En muchos casos un nodo se suele designar como la raíz del árbol. En ese caso se denomina un árbol con raíz.

Ejemplo, Ilustración

Ejemplo de Árbol Binario



Definición recursiva de árbol binario

- Un árbol vacío es un árbol binario y se representa ().
- Si A y B son dos árboles binarios y c es un nodo, entonces (A,c,B) es un árbol binario.
- No hay nada más que sea un árbol binario.

Ejemplo:

El árbol anterior se puede describir como*

(((d),b,()),a,(((g),e,()),c,(f)))

* Para simplificar la notación, se permite escribir el árbol ((),x,()) como (x)

Sucesiones descendientes

- La demostración por recursividad requiere un dominio bien fundado.
- Sea G(x,y) un predicado entre 2 miembros consecutivos del dominio. Ejemplo: x>y para x,y en $\mathbb{N}$.
- Se denomina ***sucesión G*** a una sucesión tal que G(x,y) es válida para todos los pares de miembros consecutivos de la sucesión.

Ejemplos de sucesiones descendentes

- En los naturales:

$$15, 4, 2$$

- En los reales

$$1/2,\ 1/4,\ 1/8$$

- En las expresiones SL

$$((P \wedge Q) \wedge \neg R),\ (P \wedge Q),\ P$$

Definición

Un dominio esta bien fundado con respecto a G si todas las sucesiones G concluyen.

- En todo dominio bien fundado existen elementos x tales que G(x,y) es falso para todo y. Estos elementos se llaman ***elementos mínimos***.

Principio de demostración por recursividad

Para demostrar que $\forall x P(x)$, se siguen los siguientes pasos:

- **Dominio bien fundado**: Establecer el predicado G y demostrar que todas las sucesiones G son finitas.
- **Base inductiva**: Para todo elemento mínimo, verificar $P(x)$.
- **Hipótesis**: Se toma un x arbitrario, y se asume $P(y)$ cierto para todos los y que satisfacen $G(x,y)$.
- Se demuestra $P(x)$ partiendo de $P(y)$.
- Se concluye $\forall x P(x)$.

Demostración por recursividad

- Nos da una nueva regla de inferencia
- Aplica a dominios bien fundados
- Es general: No está limitada al conjunto de los naturales.

$P(x_0)$	para x_0 elementos minimos
$\forall x \forall y\,[(G(x,y) \rightarrow P(y)) \rightarrow P(x)]$	paso inductivo
$\forall x P(x)$	Se concluye

Demostración

Se procede por contradiccion negando la conclusion $\neg\forall x P(x)$

1)	$\forall x \forall y \left[(G(x,y) \to P(y)) \to P(x) \right]$	Premisa
2)	$\neg\forall x P(x)$	Negando la conclusion
3)	$\exists x \neg P(x)$	Equiv. entre para todo y existe
4)	$\neg P(t_1)$	Particularizacion existencial. t_1 var fija
5)	$(G(t_1,t_2) \to P(t_2)) \to P(t_1)$	Particularizando $x = t_1$, $y = t_2$ en (1)
6)	$\neg\left(G(t_1,t_2) \to P(t_2)\right)$	Modus tollens (4) y (5)
7)	$\neg\left(\neg G(t_1,t_2) \vee P(t_2)\right)$	Eq. del condicional
8)	$G(t_1,t_2) \wedge \neg P(t_2)$	D'Morgan en (7)
9)	$G(t_1,t_2)$	Ley de simplificacion de (8)
10)	$\neg P(t_2)$	Ley de simplificacion de (8)

La inducción en $\mathbb{N}$ es un caso particular de la demostración por recursividad.

- El único elemento mínimo es 0.
- Se demuestra P(0).
- G(x,y) es la relación $x>y$ en $\mathbb{N}$.
- Para un x dado, se asume P(y) es cierta para todo $y<x$.
- Se demuestra P(x)

Inducción estructural

Cuando un dominio se genera por una definición recursiva sin reglas de acortamiento, el dominio siempre está bien fundado y la relación G(x,y) se entiende como x se puede generar de y (ó también como “y es un elemento propio de x”).

Ejemplos:

- Expresiones SL
- Expresiones SM
- Árboles binarios
- Listas

Inducción estructural

Para demostrar que $\forall x P(x)$

- **Base inductiva**: Se verifica P(x) para todo elemento mínimo x.
- **Hipótesis inductiva**: Se asume P(y) cierta para todos los elementos propios de x.
- **Paso inductivo**: Se demuestra P(x) partiendo de P(y).
- **Conclusión**: Se concluye $\forall x P(x)$.

Ejemplo

Se define el dual de una expresión SL así:

$$\text{dual}(A \wedge B) \equiv \text{dual}(A) \vee \text{dual}(B)$$

$$\text{dual}(A \vee B) \equiv \text{dual}(A) \wedge \text{dual}(B)$$

$$\text{dual}(\neg A) \equiv \neg\text{dual}(A)$$

$$\text{dual}(P) \equiv P, \text{ P proposición atómica}$$

Demostrar que dual(C) se sustituyen todos los $\wedge$ por $\vee$ y los $\vee$ por $\wedge$.

Ejercicio

- La negación del dual de una expresión lógica (SL) es equivalente a la expresión reemplazando todos los literales por su negación.

Ejemplo:

$\neg dual(A \vee B \vee C) \equiv \neg(A \wedge B \wedge C)$

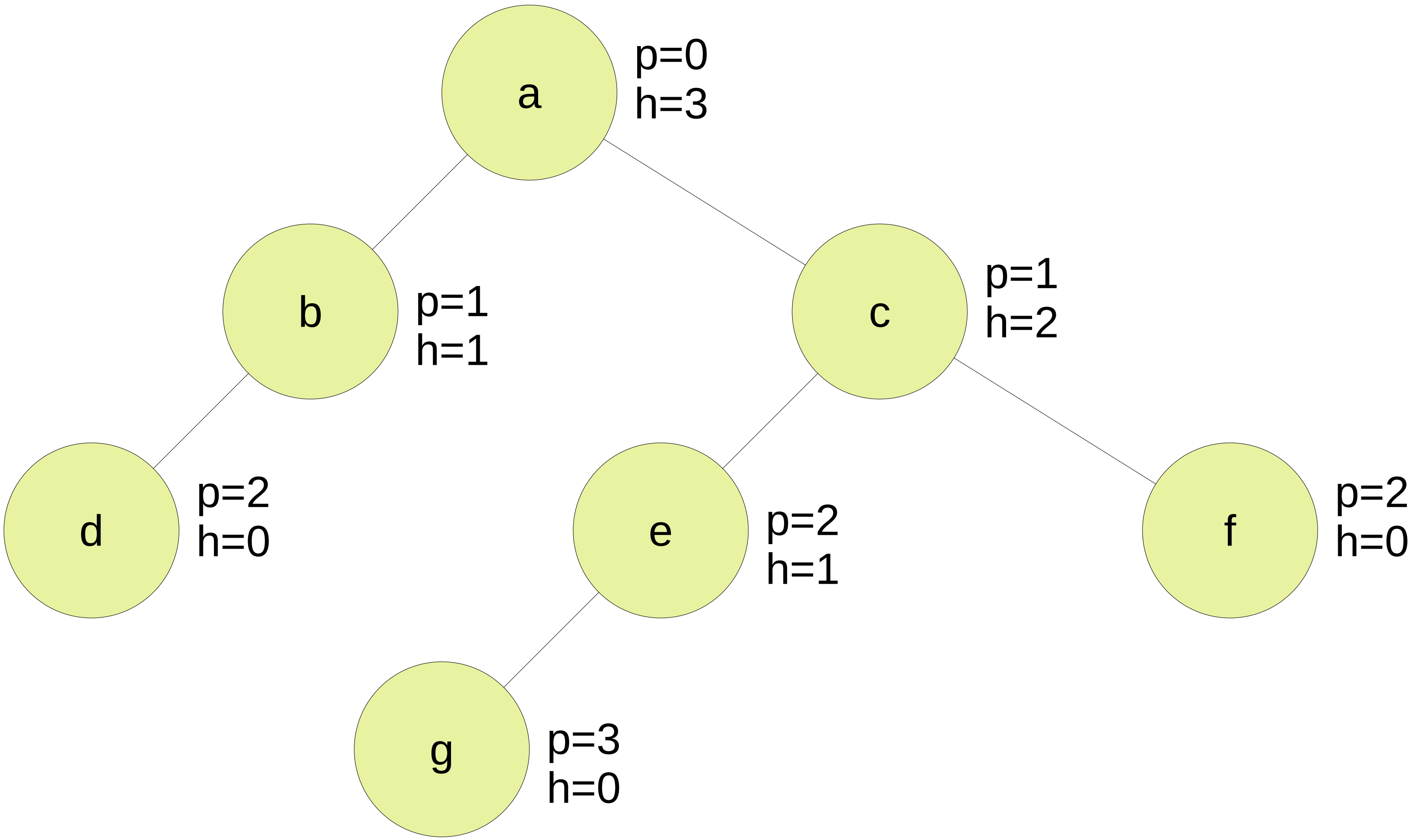
$\equiv (\neg A \vee \neg B \vee \neg C)$

Propiedades de los árboles binarios

Definiciones

- **Profundidad**: El nodo raíz tiene profundidad 0. Hijos de un nodo de profundidad p, tienen profundidad p+1.
- **Altura**: La altura de un árbol es la máxima profundidad de todos sus nodos. Las hojas del árbol tienen altura 0. Un nodo interno tiene la altura del mayor de sus hijos más uno.

Profundidad y altura



Teoremas para árboles binarios

- El máximo número de nodos que puede tener un árbol binario de altura h es $2^{h+1}-1$.
- Un árbol de altura h tiene como mínimo h+1 nodos.

Ejercicio: Demostrar estas propiedades

Más teoremas de árboles
Ejercicios

- Un árbol de n nodos tiene como máximo profundidad n-1.
- Un árbol binario con n nodos tiene como mínimo una altura(*):

$$\lceil \lg(n+1) - 1 \rceil$$