4. Hradlové sítě

(zapsal: Petr Jankovský)

Výkon počítačů nelze zvyšovat donekonečna a přestože již pěkných pár let platí, že se jejich rychlost s časem exponenciálně zvětšuje, jednou určitě narazíme přinejmenším na fyzikální limity.

Když tedy nemůžeme zvyšovat rychlost jednoho procesoru, jak počítat rychleji? Řešením by mohlo být pořídit si procesorů víc. Už dnes na běžném PéCéčku máme k dispozici vícejádrové procesory, díky nimž můžeme využít pararelní počítání a úlohu řešit tak, že práci šikovně rozdělíme mezi procesory (či jádra) a zaměstnáme je při výpočtu všechny.

My se podíváme na abstraktní výpočetní model, který je ještě paralelnější. Techniky, které si ukážeme na tomto modelu, se však dají překvapivě využít i při reálném paralelizování na několika málo procesorech. Konec konců i proto, že vnitřní architektura procesoru se našemu modelu velmi podobá. Budeme se zabývat jednoduchým modelem paralelního počítače, totiž hradlovou sítí.

Hradlové sítě

Definice: *Hradlo* je prvek, který umí vyhodnocovat nějakou funkci nad konečnou abecedou Σ.

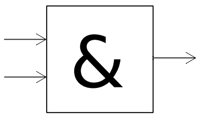
Obecně se na hradlo díváme jako na funkci $f : \Sigma^k \to \Sigma$, která dostane k vstupů a vrátí jeden výstup, přičemž hodnoty, nad kterými pracuje, budou z nějaké konečné abecedy – tedy z nějaké konečné množiny symbolů Σ. Písmenku k zde říkáme *arita hradla*.

Příklad: Často studujeme hradla booleovská (pracující nad abecedou $\{0, 1\}$), která počítají logické funkce.

Z nich nejčastěji potkáme:

- nulární: to jsou konstanty (FALSE=0, TRUE=1),
- unární: např. negace (značíme $\neg$),
- binární: logický součin (AND, &), součet (OR, $\vee$), …

Hradla kreslíme třeba následovně:

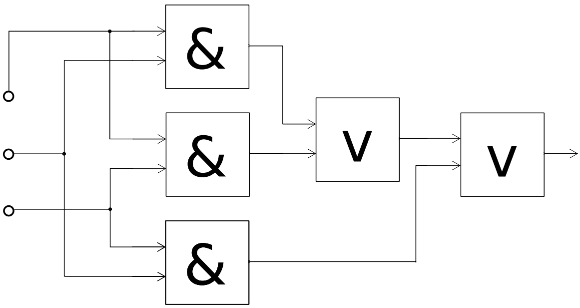


Binární hradlo provádící logickou operaci AND.

Jednotlivá hradla můžeme navzájem určitým způsobem propojovat a vytvářet z nich *hradlové sítě*. Pokud používáme pouze booleovská hradla, říkáme takto vytvořeným sítím *booleovské obvody*. Pokud pracujeme s operacemi nad nějakou obecnější (ale konečnou) množinou symbolů (abecedou), nazývají se *kombinační obvody*.

Každá hradlová síť má nějaké vstupy, nějaké výstupy a uvnitř jsou propojovaná hradla. Každý vstup hradla je připojen buďto na některý ze vstupů sítě nebo na výstup jiného hradla. Výstupy hradel mohou být propojeny na vstupy dalších hradel (mohou se větvit), nebo na výstupy sítě. Přitom máme zakázáno vytvářet cykly.

Než si řekneme formální definici, podívejme se na obrázek.



Hradlová síť – třívstupová verze funkce *majorita*.

Obrazek znázorňuje hradlovou síť, která počítá, zda je alespoň na dvou ze vstupů jednička. Pojďme si ale *hradlovou síť* definovat formálně.

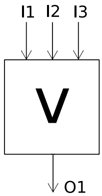
Definice: *Hradlová síť* je určena:

- *abecedou* Σ (to je nějaká konečná množina symbolů, obvykle $\Sigma = \{0, 1\}$);
- po dvou disjunktními konečnými množinami I (*vstupy*), O (*výstupy*) a H (*hradla*);
- acyklickým orientovaným multigrafem (V, E), kde $V = I \cup O \cup H$;
- zobrazením F, které každému hradlu $h \in H$ přiřadí nějakou funkci $F(h) : \Sigma^{a(h)} \rightarrow \Sigma$, což je funkce, kterou toto hradlo vykonává. Číslu $a(h)$ říkáme *arita* hradla h;
- zobrazením $z : E \rightarrow \mathbb{N}$, které každé hraně vedoucí do nějakého hradla přiřazuje některý ze vstupů tohoto hradla.

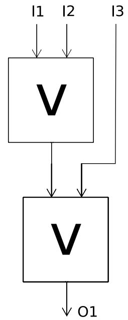
Přitom jsou splněny následující podmínky:

- $\forall i \in I : \deg^{in}(i) = 0$ (do vstupů nic nevede);
- $\forall o \in O : \deg^{in}(o) = 1 \,\&\, \deg^{out}(o) = 0$ (z výstupů nic nevede a do každého vede právě jedna hrana);
- $\forall h \in H : \deg^{in}(v) = a(v)$ (do každého hradla vede tolik hran, kolik je jeho arita);
- $\forall h \in H \; \forall j : 1 \leq j \leq a(h)$ existuje právě jedena hrana e taková, že e končí v h a $z(e) = j$, (všechny vstupy hradel jsou zapojeny).

Pozorování: Kdybychom připustili hradla s libovolně vysokým počtem vstupů, mohli bychom libovolný problém se vstupem délky n vyřešit jedním hradlem o n vstupech, kterému bychom přiřadili funkci, která by naši úlohu rovnou vyřešila. Tento model však není ani realistický, ani pěkný. Proto přijměme omezení, že arity všech hradel budou omezeny nějakou pevnou konstantou k (ukáže se, že nám bude stačit dvojka a vystačíme si tedy pouze s nulárními, unárními a binárními hradly). Následující obrázky ukazují, jak hradla o více vstupech nahradit dvouvstupovými:



Trojvstupové hradlo OR.

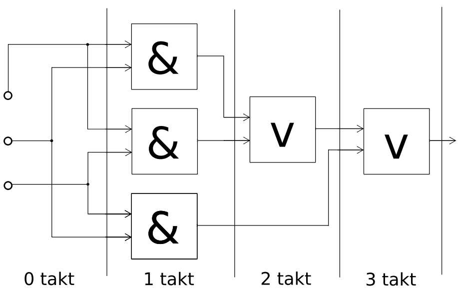


Jeho nahrazení 2-vstupovými hradly.

Nyní bychom ještě měli definovat, co taková hradlová síť vlastně počítá a jak její výpočet probíhá.

Definice: *Výpočet sítě* probíhá po *taktech.* V nultém taktu jsou definovány pouze hodnoty na vstupech sítě a na výstupech hradel arity 0. Můžeme si to představit tak, že na začátku nemá žádné hradlo definovánu výstupní hodnotu (až na již zmíněná hradla nulární). V každém dalším taktu pak vydají výstup hradla, která na konci minulého taktu měla definovány všechny hodnoty na vstupech. Jakmile budou po nějakém konečném počtu taktů definované i hodnoty všech výstupů, síť se zastaví a vydá výsledek.

Pozorování: Protože je síť acyklická, je jasné, že jakmile jednou nějaké hradlo vydá výstup, tak se tento výstup během dalšího výpočtu sítě již nezmění.



Výpočet hradlové sítě.

Podle toho, jak síť počítá, si ji můžeme rozdělit na vrstvy:

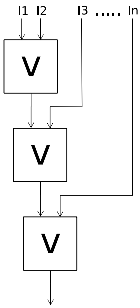
Definice: *i-tá vrstva S_i* obsahuje právě takové vrcholy v, pro které nejdelší cesta ze vstupů sítě do v má délku rovnou i.

Pozorování: Všimněme si, že v i-tém taktu vydají hodnoty právě hradla z S_i.

Dává tedy smysl prohlásit za *časovou složitost* sítě počet jejích vrstev. Podobně *prostorovou složitost* definujeme jako počet hradel v síti.

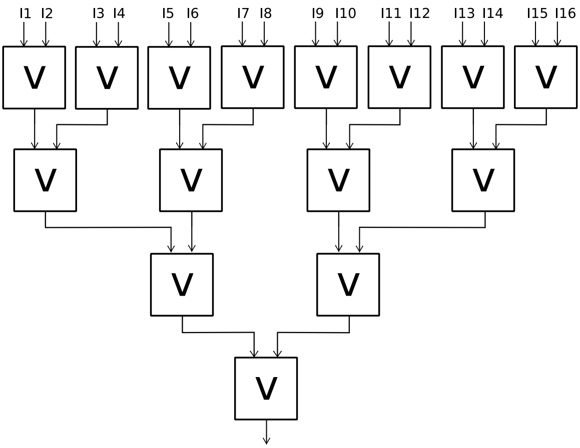
Příklad: Sestrojme síť, která zjistí, zda se mezi jejími n vstupy vyskytuje alespoň jedna jednička.

První řešení: zapojíme hradla za sebe (sériově). Časová i prostorová složitost odpovídají $\Theta(n)$. Zde ovšem vůbec nevyužíváme toho, že by mohlo počítat více hradel současně.



Hradlová síť, která zjistí, zdali je na vstupu alespoň jedna jednička.

Druhé řešení: Hradla budeme spojovat do dvojic, pak výsledky z těchto dvojic opět do dvojic a tak dále. Díky paralelnímu zapojení dosáhneme časové složitosti $\Theta(\log n)$, prostorová složitost zůstane lineární.



Chytřejší řešení stejného problému pro vstup velikosti 16.