In fact, we can show that a CSP which is 1-consistent need not be 1-satisfiable. This would be the case if there exist some variables which have empty domains, and all the values in the nonempty domains satisfy the constraints of the corresponding variables. Theorem 3-3 states that a CSP which has all the domains and constraints as empty sets is strong k-consistent for all k.

Theorem 3-3

A CSP in which all the domains are empty sets is strong k-consistent for all k:

$$\forall \text{ csp}((Z, D, C)) \Rightarrow$$
$$(\forall D_x \in D: D_x = \{\}) \Rightarrow (\forall k \leq |Z|: \text{strong } k\text{-consistent}((Z, D, C))))$$

Proof

Let $\mathcal{P} = (Z, D, C)$ be a CSP in which all the domains are empty sets. It is 1-unsatisfiable by definition. It is also h-unsatisfiable for all $1 \leq h \leq |Z|$ because no h-compound label h-satisfies C. However, $\mathcal{P}$ is 1-consistent (by definition of 1-consistency, since for all x, D_x is empty). For any $k > 1$, there exists no $(k-1)$-compound label which $(k-1)$-satisfies the constraints of $\mathcal{P}$, and therefore the left hand side of the "$\Rightarrow$" in the definition of k-consistency (Definition 3-4) is never satisfied. Therefore, the proposition k-consistency($\mathcal{P}$) is always true for all k, which means strong k-consistency($\mathcal{P}$) is always true.

(Q.E.D.)

One significant implication of Theorem 3-3 is that strong n-consistency itself does not guarantee n-satisfiability. Careful analysis shows that 1-satisfiability together with strong k-consistency is a sufficient (but not necessary) condition to k-satisfiability.

Theorem 3-4 (The Satisfiability Theorem)

A CSP which is 1-satisfiable and strong k-consistent is k-satisfiable for all k:

$$\forall \text{ csp}(\mathcal{P}): \text{1-satisfiable}(\mathcal{P}) \wedge \text{strong } k\text{-consistent}(\mathcal{P}) \Rightarrow k\text{-satisfiable}(\mathcal{P})$$

Proof

Let $\mathcal{P} = (Z, D, C)$ be 1-satisfiable and strong k-consistent for some integer k. Pick an arbitrary subset of k variables $S = \{z_1, z_2, ..., z_k\}$ from Z. We shall prove that there exists at least one compound label for all the variables in S which satisfies all the relevant constraints (i.e. CE(S, $\mathcal{P}$)).

Since $\mathcal{P}$ is 1-satisfiable, for any arbitrary element x_1 that we pick from S, we can at least find one value v_1 from the domain of x_1 such that satisfies($<x_1,v_1>$, C_{x_1}) holds. Furthermore, since $\mathcal{P}$ is 2-consistent, for any other variable x_2 that we pick from S, we would be able to find a compound label ($<x_1, v_1><x_2, v_2>$) which satisfies CE($\{x_1,x_2\}$, $\mathcal{P}$). Since $\mathcal{P}$ is strong-k-consistent, it should not be difficult to show by induction that for any 3rd, 4th, ..., kth variables in S that we pick, we shall be able to find 3-, 4-, ..., k-compound labels that satisfy the corresponding constraints CE($\{x_1, x_2, ..., x_k\}$, $\mathcal{P}$). Therefore, the subproblem on S is satisfiable, and so $\mathcal{P}$ is k-satisfiable.

(Q.E.D.)

We summarize below the results that we have concluded so far:

(1) k-satisfiability subsumes $(k-1)$-satisfiability (trivial).

(2) However, k-consistency does not entail $(k-1)$-consistency. This is illustrated by example CSP-1, which is 3-consistent but not 2-consistent. But some k-consistent CSPs must be $(k-1)$-consistent, and *vice versa*. This leads to the definition of strong k-consistency, which entails strong $(k-1)$-consistency.

(3) k-consistency does not guarantee 1-satisfiability. Consequently, k-consistency does not guarantee h-satisfiability for any h. This is true for $k \leq h$, as illustrated in the example CSP-2 which is 3-consistent but not 4-satisfiable. It is also true for $k > h$, as it is illustrated by the colouring problem CSP-4 in Figure 3.4, which is 3-consistent, but not 2-satisfiable.

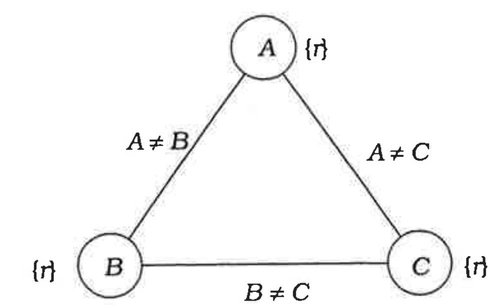

Figure 3.4 CSP-4: a CSP which is 1 satisfiable and 3-consistent, but 2-inconsistent and 2-unsatisfiable (it is 3-consistent because there is no 2-compound label which satisfies any of the binary constraints)

3.3 Relating Consistency to Satisfiability

Before we continue, let us examine the relationship between the satisfiability and consistency concepts that we have introduced so far. In particular, is *k*-consistency, or strong *k*-consistency, a sufficient or necessary condition for *k*-satisfiability? Is *k*-consistency, or strong *k*-consistency, a sufficient or necessary condition for the satisfiability of a problem? These questions will be answered in this section.

It is not difficult to show that *k*-consistency is insufficient to guarantee satisfiability of a CSP which has more than *k* variables. For example, the colouring problem CSP-2 shown in Figure 3.2 is a 3-consistent but unsatisfiable CSP.

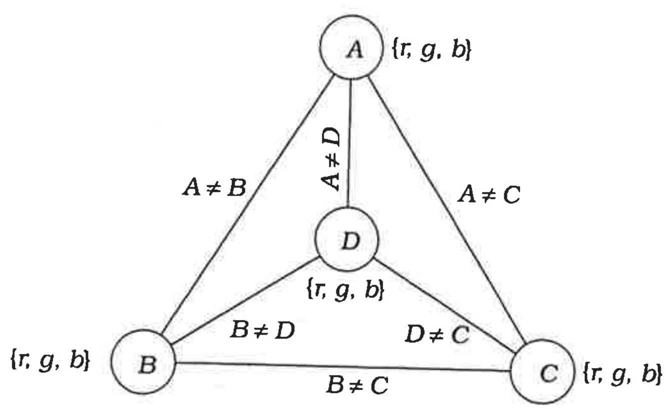

Figure 3.2 CSP-2: example of a 3-consistent but unsatisfiable CSP constraint: no adjacent nodes should take the same value (from Freuder, 1978)

The domains of the variables are shown in curly brackets next to the variables in Figure 3.2. On the edges, the compound labels allowed for the joined nodes are shown. CSP-2 is 3-consistent because whatever combination of three variables that we pick, assigning two of them any two different values from "*r*", "*g*" and "*b*" would allow one to assign the remaining value to the remaining variable without violating any of the constraints on the three variables. But this problem is unsatisfiable because one needs four values to label all the variables without having any adjacent variables taking the same value.

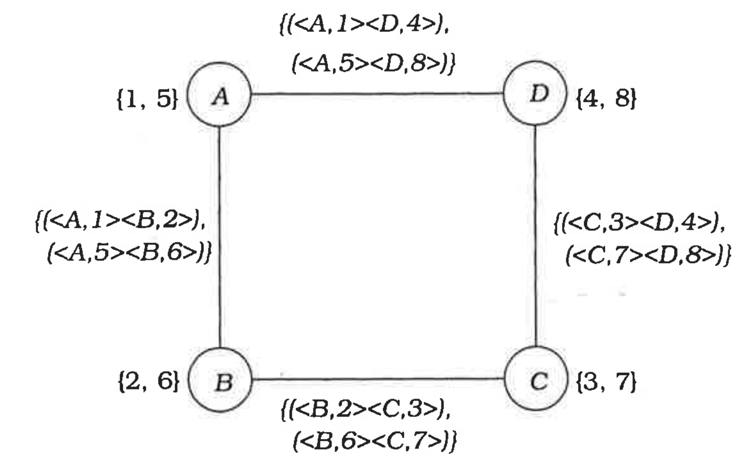

Figure 3.3 CSP-3: a problem which is satisfiable but not path-consistent. The variables are *A*, *B*, *C* and *D*; their domains are shown next to the nodes which represent them. The labels on the edges show the sets of all compatible relations between the variables of the adjacent nodes

The example CSP-3 in Figure 3.3 shows that 3-consistency is not a *necessary condition* for satisfiability either. In CSP-3, if $A = 1$, then from $C_{A,B}$ we have to make $B = 2$, which by $C_{B,C}$ forces $C = 3$, which by $C_{C,D}$ forces $D = 4$. Similarly, if $A = 5$, then $B = 6$, which forces $C = 7$, which in turn forces $D = 8$. Therefore, two and only two compound labels for the variables in the problem satisfy all the constraints:

(<*A*, 1><*B*, 2><*C*, 3><*D*, 4>)

and (<*A*, 5><*B*, 6><*C*, 7><*D*, 8>)

But consider the compound label (<*A*,1><*C*,7>): it satisfies all the constraints C_A, C_C and $C_{A,C}$ ($C_{A,C}$ is not a constraint stated in the problem, and therefore not shown in Figure 3.3). But no value for *B* is compatible with (<*A*,1><*C*,7>) (<*B*,2> violates the constraint $C_{B,C}$ and <*B*,6> violates the constraint $C_{A,B}$). Therefore PC((*A*, *B*, *C*), CSP-3) is false; in other words, PC does not hold for CSP-3. This example shows that path-consistency, or 3-consistency, is not a necessary condition for satisfiability of a CSP. Therefore, *k*-consistency is neither a necessary nor a sufficient condition for satisfiability.